UNIVERSITY OF CALIFORNIA,
IRVINE

The Efficient Discrimination of Electron and Pi-Zero Events in a Water Cherenkov Detector and the Application to Neutrino Oscillation Experiments

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Physics

by

Tomasz Barszczak

Dissertation Committee:
Professor Henry W. Sobel, Chair
Professor David Casper
Professor Jonas Schultz
Professor Dennis Silverman

2005

The dissertation of Tomasz Barszczak
is approved and is acceptable in quality
and form for publication on microfilm:

Committee Chair

University of California, Irvine
2005

DEDICATION

To Tao, Adam, Izabela, and Daniel.

Table of Contents

List of Figures

List of Tables

ACKNOWLEDGEMENTS

I would like to express special thanks to my advisor, Professor Henry Sobel, for providing an excellent working environment, to David Casper, for providing an invaluable guidance, especially on fitter development, to Danuta Kiełczewska, who introduced me to the Super-Kamiokande experiment, to Shunichi Mine, who, in my absence, added the neutral pion fitter to the official ntuple generation, to Ed Kearns, for providing guidance, especially at the final stages of this work, and to all the members of the experiment, whose contributions made this work possible.

This work has been supported by the United States Department of Energy as well as the University of California, Irvine.

CURRICULUM VITAE

Tomasz Barszczak

1993	B.S. in Physics, Warsaw University, Poland.
1997	M.S. in Physics, University of California, Irvine.
2005	Ph.D. in Physics, University of California, Irvine.

ABSTRACT OF THE DISSERTATION

The Efficient Discrimination of Electron and Pi-Zero Events in a Water Cherenkov Detector and the Application to Neutrino Oscillation Experiments

By

Tomasz Barszczak

Doctor of Philosophy in Physics

University of California, Irvine, 2005

Professor Henry W. Sobel, Chair

Super-Kamiokande, a large water Cherenkov detector, observed atmospheric neutrinos produced by interactions of cosmic rays in the atmosphere. By observing zenith angle and energy of the particles produced by the neutrinos in single ring events in Super-Kamiokande, it became apparent that the muon neutrino (ν_μ) undergoes oscillations. During the oscillations, ν_μ changes into another kind of neutrino, which can be either the tau neutrino (ν_τ) or a sterile neutrino (ν_s). In addition, a small component of ν_e is possible but not considered here.

With the standard single ring analysis, using two-flavor oscillations without matter effects, it was hard to discriminate between these two possibilities because, while ν_s doesn't interact at all and ν_τ does, interactions of ν_τ don't produce a significant single ring signal. Due to the large mass of the taon (τ), charged current interactions of ν_τ are suppressed in the observed energy region. However, it is possible to study neutral current interactions in order to discriminate between the two oscillation possibilities.

This work describes a technique for extracting the neutral current interactions from Super-Kamiokande data by means of selecting $\pi^0 \rightarrow \gamma\gamma$ two ring events. Additionally, two ring events are used to independently confirm the neutrino oscillation observation.

A crucial step in the selection of neutral pion events is the reconstruction of the two Cherenkov rings produced in π^0 decays. Described here is a new reconstruction algorithm that uses a maximum likelihood method to fit the expected pattern of light to the observed pattern of light. A fast method is used for generation of the light pattern (including scattered light) produced by a hypothetical particle track configuration. Also described here are software components created in order to develop the fitter, including a custom data visualization program, and inter-language interface.

A new generation of long baseline neutrino oscillation experiments is coming up. An important goal of these experiments is search for electron neutrino (ν_e) appearance. Neutral pions are an important background. This algorithm is a valuable tool for discrimination between electrons produced by ν_e and background neutral pions in experiments such as K2K and T2K.

Chapter 1

Introduction

1.1 History of the Neutrino

The neutrino [1, 2] is an elementary particle first postulated by Wolfgang Pauli in order to save the laws of conservation of energy and momentum. Electrons that are emitted in radioactive beta decay have a continuous energy spectrum. However, if the decaying nucleus emits only an electron, and nothing else, the electron should have a monochromatic energy spectrum, as in any two body decay process. Pauli assumed that another light, neutral particle is emitted simultaneously from the nucleus during the beta decay. It subsequently escapes undetected.

On the 4th of December 1930, in a letter to the radioactive group at a regional meeting in Tuebingen, he wrote [3]:

> I have hit on a desperate way to escape from the problems of the "wrong" statistics of the N and Li^6 nuclei and of the continuous beta spectrum in

> order to save the “even-odd” rule of statistics and the law of conservation of energy. Namely the possibility that electrically neutral particles, which I would like to call neutrons might exist inside nuclei; these would have spin 1/2, would obey the exclusion principle, and would in addition differ from photons through the fact that they would not travel at the speed of light. The mass of the neutron ought to be about the same order of magnitude as the electron mass, and in any case could not be greater than 0.01 proton masses. The continuous beta spectrum would then become understandable by assuming that in beta decay a neutron is always emitted along with the electron, in such a way that the sum of the energies of the neutron and electron is a constant.

This was a radical idea at that time. Pauli called the new particle neutron (German neutronen), following the naming convention of the other particles known at that time—electron and proton.

In February 1932, J. Chadwick discovered the neutron [4], but it was too heavy to be the particle postulated by Pauli.

In 1934 Enrico Fermi developed a theory of beta decay [5] and gave the neutrino its current name, which means “little neutral one” in Italian. He explained that beta decay is, in effect, a decay of the neutron to a proton, electron, and neutrino. In contemporary notation, we write:

$$n \to p + e^- + \overline{\nu}_e \, . \tag{1.1}$$

His theory correctly explained the spectrum of electrons emitted in beta decay.

Soon, people realized that neutrinos have a very low interaction cross-section [6]. A single neutrino of moderate energy is able to pass unhindered through a wall of lead of a thickness measured in light-years. Pauli worried that he had done “a frightful thing", as he called it, by proposing a particle that could never be discovered. It took almost three decades from the time the neutrino was first postulated to see it experimentally.

The neutrino was first observed in 1953-1956 by Frederick Reines and Clyde L. Cowan [7, 8]. They used a nuclear reactor as a powerful source of neutrinos. A series of experiments, first at Hanford, and then at Savannah River nuclear plants, suffered at first from low counting rates and high backgrounds. Initial results were not definite but, as the experiments improved, they showed a convincing signal.

The key to the success was a delayed double coincidence measurement. The target consisted of 400 liters of solution of cadmium chloride in water. An antineutrino from the reactor reacted with a proton in a water molecule giving a positron and a neutron:

$$\overline{\nu}_e + p \to n + e^+ \ . \tag{1.2}$$

The positron annihilated with an electron giving two simultaneous photons. The neutron slowed down and was captured by the cadmium nucleus, which subsequently emitted photons, about 15 microseconds after the positron annihilation. The photons from the positron annihilation and from the neutron capture were detected by an array of photomultipliers surrounding the target. The detection of the sequence of photons from the position followed by the photons from the neutron capture, in a delayed coincidence fashion, reduced the background sufficiently. In 1995, Frederick Reines received the Nobel Prize for the discovery.

In 1956, Tsung-Dao Lee and Chen-Ning Yang, while trying to understand decays of strange particles, proposed that weak interactions violate spatial parity symmetry [9]. A few months later, Abdus Salam realized that such parity violation could be explained if the neutrino was massless [10].

In December that year, Chien-Shiung Wu observed parity non-conservation. She looked at the angular distribution of electrons emitted from beta decay of cobalt-60 cooled to a fraction of a degree above absolute zero and placed in a magnetic field. Most of the electrons were emitted in the direction opposite to the direction of the magnetic field. In 1957, Maurice Goldhaber determined that neutrinos are left-handed, by observing polarization of gamma rays emitted after electron capture by Europium isotope Eu^{152m}.

In 1959, Pontecorvo proposed that there exists a second kind of neutrino. Physicists observed decays of muons into an electron and two neutrinos. Today we write:

$$\mu^- \to e^- \overline{\nu}_e \nu_\mu \ . \tag{1.3}$$

But it seemed that the muon never decays into an electron and a photon, i.e., that the reaction

$$\mu^- \nrightarrow e^- \gamma \tag{1.4}$$

does not occur. Treating lepton numbers for electrons and muons separately explains why this reaction doesn't happen. This also implies that the two neutrinos produced in muon decay are of different kinds.

In 1962, Leon Lederman, Jack Steinberger, and Melvin Schwartz observed that neutrinos produced in pion decays

$$\pi^+ \rightarrow \mu^+\nu_\mu \tag{1.5}$$

are different from neutrinos produced in beta decays. In the first high energy neutrino experiment at Brookhaven National Laboratory, they directed a proton beam on a target to produce pions which decayed in flight. After a short distance, all particles, except neutrinos, were stopped by steel and concrete. They used a 10 t spark chamber surrounded by a cosmic ray veto and scintillator trigger counters. The neutrino type was identified by the type of particle it produced: an electron, or a muon. Their observation showed that all neutrinos were of muon type, and a small number of observed electron events was consistent with an expected background level. In 1988, they received the Nobel Prize for their discovery.

After the two families of leptons were established, there was a question whether more lepton families exist. Martin Perl looked for a third lepton at SPEAR (Stanford Position Electron Asymmetric Collider) at SLAC. He knew that the third lepton had to be heavy, otherwise it would have been observed previously. Because of its expected large mass, it could decay into pions, which don't give a unique detection signature. Instead, he looked at events in which two oppositely charged leptons decay into an electron and muon. For example, today we write:

$$\tau^+ \rightarrow \mu^+\overline{\nu}_\tau\nu_\mu \tag{1.6}$$

$$\tau^- \rightarrow e^-\nu_\tau\overline{\nu}_e\ . \tag{1.7}$$

In 1974 he observed such events. He called the new particle tau (τ), from the first letter of word "third" in Greek. For this discovery, he received the Nobel Prize in 1995.

There exists a third neutrino associated with the tau particle, ν_τ. It was directly observed in 2000 in the DONUT experiment at Fermilab [11].

In 1960's, Abdus Salam and Steven Weinberg independently developed an idea of the unification of weak and electromagnetic interactions into one electroweak force (Sheldon Glashow extended this idea from leptons to hadrons). In 1971, Gerard 't Hooft managed to make the electroweak theory renormalizable, which caused it to be considered more seriously. The electroweak theory predicted the existence of a neutral gauge boson Z^0, which mediates neutral current interactions. In neutral current interactions of neutrinos, the neutrino scatters elastically off an electron or a nucleus, transferring momentum to it, without producing a corresponding lepton (a muon in case of ν_μ, or an electron in case of ν_e). In 1973, a team at the Gargamelle bubble chamber detector at CERN observed neutral current interactions of neutrinos.

In 1984, the UA1 detector at CERN observed the W boson directly. The discovery was soon followed by the observation of the Z^0.

In 1980s, a new machine was built at CERN: the LEP—Large Electron-Positron collider. It was 27 km in circumference and it was capable of producing Z^0 particles in large quantities from electron-positron annihilation. Z^0 decays into quark-antiquark or lepton-antilepton pairs, including neutrino-antineutrino pairs. The neutrino pairs are not directly observable, but they contribute to the total width of the Z^0. By measuring this width, LEP experiments determined that there are exactly three kinds of light neutrinos. The measured number of neutrinos is 2.994 ± 0.012 [12].

1.2 Neutrino Mass

Soon after the neutrino was postulated by Pauli, it became obvious that its mass is very small, much smaller than the mass of an electron in the case of ν_e. In fact, parity violation suggests that neutrinos are massless. So far, all direct neutrino mass measurements are only upper limits.

The electron neutrino mass is usually determined by measuring the shape of the end point of spectrum of electrons from tritium beta decay. The measurement is complicated by the fact, that in earlier tritium experiments, pileup of events at the spectrum endpoint due to an unknown physical effect caused the fit value of $m^2_{\nu_e}$ to be negative. This has been fixed in the latest experiments, and the best current limit is $m_{\nu_e} < 3$ eV [12, 13, 14].

The Mass of the muon neutrino is determined from the muon spectrum from pion decays at rest:

$$\pi^+ \to \mu^+ \nu_\mu \ . \tag{1.8}$$

The best current limit is $m_{\nu_\mu} < 0.19$ MeV (90% CL) [12, 15].

The mass of the tau neutrino is determined from kinematics of

$$\tau^- \rightarrow 2\pi^-\pi^+\nu_\tau \tag{1.9}$$

$$\tau^- \rightarrow 3\pi^-2\pi^+(\pi^0)\nu_\tau \tag{1.10}$$

decays. The best current limit is $m_{\nu_\tau} < 18.2$ MeV (95% CL) [16].

There are also less conventional methods of measurement of neutrino masses. In 1987, neutrinos from supernova SN1987, that exploded in the Large Magellanic Cloud, were observed in the Kamiokande and IMB detectors. The spread of arrival times of the neutrinos, coupled with the measured neutrino energies, provided a time of flight limit on neutrino mass. That limit, however, is no longer competitive with the limits from tritium beta decay. Searches for neutrinoless double beta decay, that violate total lepton number conservation, provide limits on the Majorana neutrino mass of the order of few eV [12].

Another limit is derived from the analysis of the shape of cosmic microwave background radiation from Wilkinson Microwave Anisotropy Probe (WMAP), CBI, and ACBAR [17]. Combined with large scale structure data from Two Degree Field (2dF) Galaxy Redshift Survey, and, optionally, with Lyman α forest data, we get a constraint that the sum of the neutrino masses is below 0.7 eV.

The interpretation of the limits on neutrino masses of neutrinos of specific flavor changes somewhat when we consider neutrino mixing and oscillations.

1.3 Neutrino Mixing and Oscillations

Since the 1960's, we have observed a deficit of neutrinos coming from the Sun [18]. A proposed solution to the solar neutrino problem was neutrino oscillations.

If neutrinos have masses, then, in general, a neutrino produced in a weak interaction associated with a charged lepton of a given flavor α is the state

$$|\nu_\alpha\rangle = \sum_i U^*_{\alpha i} |\nu_i\rangle \ , \tag{1.11}$$

where i indexes neutrino mass eigenstates, and U is the unitary lepton mixing matrix [19]. When such a neutrino of a definite flavor propagates in space, mass eigenstates propagate with different rates of phase change

$$|\nu_i(L)\rangle \approx e^{-i(m_i^2/2E)L} |\nu_i(0)\rangle \ , \tag{1.12}$$

where L is distance, E is energy, m_i is mass, and we assumed that the neutrino is very relativistic, i.e., its energy is much greater than its mass. After traveling a distance L, the superposition of the mass eigenstates produces a state that consists of a mixture of flavors. The probability of detection of flavor β is

$$\begin{aligned} P(\nu_\alpha \to \nu_\beta) = \delta_{\alpha\beta} - & \\ & 4 \sum_{i>j} \Re(U^*_{\alpha i} U_{\beta i} U_{\alpha j} U^*_{\beta j}) \sin^2[1.27\,\Delta m^2_{ij}(L/E)] + \\ & 2 \sum_{i>j} \Im(U^*_{\alpha i} U_{\beta i} U_{\alpha j} U^*_{\beta j}) \sin[2.54\,\Delta m^2_{ij}(L/E)] \ , \end{aligned} \tag{1.13}$$

where $\Delta m^2_{ij} \equiv m_i^2 - m_j^2$, mass is in eV, L is in km, and E is in GeV.

In many experimental situations, we can use a two flavor approximation that simplifies equation 1.13 considerably. One such case is when one mass splitting is much larger than the others, and L/E involved in the experiment is smaller than the inverse of the remaining mass splittings [20]. Another case is when only two flavors are relevant; for example, when only two mass eigenstates couple strongly with the studied flavor. When only two eigenstates count, we can write the unitary mixing matrix as

$$U = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} , \tag{1.14}$$

where θ is called the mixing angle. The probability of detection of a neutrino flavor different than the produced flavor becomes

$$P = \sin^2 2\theta \, \sin^2[1.27\,\Delta m^2 (L/E)] \, . \tag{1.15}$$

The probability of detection of the produced flavor is $1 - P$. The phenomenon of neutrino flavor change is referred to as oscillations because of the $\sin^2(L/E)$ probability distribution.

When neutrinos travel through matter (for example, the Sun or Earth), their coherent forward scattering from particles they encounter along the way can significantly modify their propagation [21]. As a result, the probability for changing flavor can be significantly different than it is in vacuum [22]. The matter effect comes into play when such scattering differs for neutrino flavors involved. For example, W-exchange-induced ν_e forward scattering from ambient electrons influences $\nu_e \leftrightarrow \nu_\mu$ and $\nu_e \leftrightarrow \nu_\tau$

oscillations. Z-exchange-induced forward scattering doesn't contribute, because it is the same for the three neutrino flavors. Matter doesn't influence $\nu_\mu \leftrightarrow \nu_\tau$ oscillations. If there were one or more additional neutrino flavors that don't interact through neutral current or charged current (more on this later), the matter effect would influence its oscillations to the three ordinary neutrino flavors.

The sun produces only electron neutrinos. Neutrino oscillations of ν_e to another type of neutrino can explain the observed deficit. Solar neutrinos were first observed using radiochemical experiments, chlorine based (Davis's pioneering experiment at Homestake [18]) and then gallium based. They were later joined by water Cherenkov experiments, Kamiokande, Super-Kamiokande [23], and the Sudbury Neutrino Observatory (SNO) [24]. These experiments perform real-time observations and provide directional information, however, they have higher detection energy threshold. All these experiments, taken together, favor the so-called LMA (large mixing angle) solution with matter-induced oscillations around $\Delta m^2 = 5.5 \times 10^{-5}\mathrm{eV}^2$ and $\tan^2\theta = 0.42$ [25]. Recently, SNO measured the total flux of the three neutrino flavors taken together. This measurement agrees with predictions, thus indicating that electron neutrinos change flavor to some combination of the remaining two neutrino types [26].

Various reactor and accelerator experiments also searched for ν_e oscillations. They reported negative results for higher Δm^2 values. Recently, the KamLAND experiment in Kamioka obtained a positive result observing neutrino oscillations from nuclear reactors in Japan, that also points to the LMA solution [27] ($\Delta m^2 = 7.9^{+0.6}_{-0.5} \times 10^{-5}\mathrm{eV}^2$

and $\tan^2\theta = 0.40^{+0.10}_{-0.07}$ [28]).

Another indication of neutrino oscillations came from the observations of atmospheric neutrinos. Atmospheric neutrinos are the decay product of hadronic showers produced by cosmic ray interactions in the atmosphere. Most neutrinos come from the charged pion decay chain:

$$\pi \rightarrow \mu\nu_\mu \tag{1.16}$$

$$\mu \rightarrow e\nu_e\nu_\mu \; . \tag{1.17}$$

This process produces neutrinos in the ratio of $\nu_\mu : \nu_e = 2 : 1$. The absolute neutrino flux cannot be precisely calculated; it is known with about 20% accuracy. However, the ratio of neutrino flavors is predicted to be known with better than 5% accuracy. Experiments have observed a ratio which is too low, i.e., too few muon neutrinos compared to electron neutrinos.

More recently, Super-Kamiokande has observed an asymmetry of the zenith angle distribution of the muon neutrino flux. Whereas the predicted flux is symmetric, we observed too few muon neutrinos going upwards. The upward-going neutrinos traveled a large distance, of the order of ten thousand kilometers from their point of production to detection, while the downward-going neutrinos traveled at most tens of kilometers. This observation is interpreted as a distance dependent oscillation of one neutrino flavor into another. The negative result of short baseline reactor neutrino limits, and the lack of distortion of the electron neutrino distribution, indicates that the muon neutrino does not oscillate to the electron neutrino, except possibly a small fraction of the time. This implies that ν_μ oscillates to ν_τ, if we neglect the possibility of another kind of "sterile" neutrino (more on this later). The allowed range at 90% CL is [29]:

$$1.5 \times 10^{-3}\mathrm{eV}^2 \lesssim \Delta m^2 \lesssim 3.4 \times 10^{-3}\mathrm{eV}^2 \ , \tag{1.18}$$

and

$$\sin^2 2\theta > 0.92 \ . \tag{1.19}$$

Recently, an accelerator experiment from KEK to Super-Kamiokande (K2K) observed a deficiency and spectral shape distortion of ν_μ produced 250 km away, that is consistent with the atmospheric neutrino result [30, 31].

The Liquid Scintillator Neutrino Detector (LSND) experiment studied muon decays at rest:

$$\mu^+ \to e^+ \nu_e \overline{\nu}_\mu \ . \tag{1.20}$$

While this decay doesn't produce $\overline{\nu}_e$, that experiment reported an excess of electron antineutrinos over background. This excess is interpreted as due to oscillation of some $\overline{\nu}_\mu$ produced in μ^+ decay into $\overline{\nu}_e$ [32]. The KArlsruhe Rutherford Medium Energy Neutrino (KARMEN) experiment has ruled out part of the parameter space indicated by LSND [33]. The remaining allowed regions are [34] $0.2 \lesssim \Delta m^2 \lesssim 1\mathrm{eV}^2$ and $0.003 \lesssim \sin^2\theta \lesssim 0.03$, or $\Delta m^2 \simeq 7\mathrm{eV}^2$ and $\sin^2\theta \simeq 0.004$. MiniBooNE experiment at Fermilab, that is currently taking data, covers the entire above allowed range, and will definitely confirm or refute LSND result [35].

The three different allowed regions: the solar neutrino range (including KamLAND), the atmospheric neutrino range (including K2K), and the results of the LSND experiment cannot be reconciled within the framework of three neutrino generations. The squared mass differences must satisfy

$$\Delta m_{32}^2 + \Delta m_{21}^2 + \Delta m_{13}^2 = 0 \; . \tag{1.21}$$

However, the three squared mass difference ranges are of three different orders of magnitude, and, therefore, cannot satisfy the constraint of equation 1.21. If all observations are correct, nature must contain at least four types of neutrinos. Since the LEP result limits the number of neutrinos to three [12], the fourth kind of neutrino would have to be a so-called sterile neutrino (ν_s), which does not interact by means of weak interactions, and, therefore, LEP wouldn't be sensitive to it. The sterile neutrino would not interact by means of any known interactions except gravity. However, it would be accessible to study by means of neutrino oscillations.

Super-Kamiokande observations can be interpreted as either $\nu_\mu \leftrightarrow \nu_\tau$ or $\nu_\mu \leftrightarrow \nu_s$ oscillations. One way to distinguish between these two possibilities is to look for matter effect on the oscillations. Because ν_s coherent forward scattering off matter particles differs from that of ν_μ (it doesn't occur), the presence of matter modifies the flavor change probability. This doesn't happen in the ν_τ case, because such scattering is identical for ν_μ and ν_τ. Matter effect can suppress flavor change probability for high energy neutrinos.

Another way to distinguish between these two possibilities, is the observation of neutral current (NC) events. Due to the high threshold of τ production and the lack of a clean signature of τ events, the charged current interactions of tau neutrino are not directly observable in Super-Kamiokande. But the ν_τ interacts through neutral current interactions while ν_s does not. If a muon neutrino transforms into a tau neutrino we would see no change in the rate of upward-going neutral current interactions. If it transforms into a sterile neutrino there would be a deficit of upward-going NC events.

In charged current interactions, the neutrino produces a lepton corresponding to the neutrino flavor. We can select neutral current interactions by looking for events in which only a π^0 is produced. The π^0 immediately decays into two photons:

$$\pi^0 \to \gamma\gamma \; . \tag{1.22}$$

The gamma particles create electromagnetic showers which produce two fuzzy ("electron-like") Cherenkov rings (see Section 5.3).

If ν_τ oscillates to ν_s there would be a deficit of upward-going π^0 events. This can be observed in two ways. There would be an asymmetry in the zenith angle distribution of π^0 events. Also, there would be a reduction of the overall π^0 event rate. However, each of those methods has some difficulties. π^0 events observed in Super-Kamiokande are most abundant at relatively low energies, where directional correlation between the pion and the neutrino is poor. We are forced to select only the less frequent higher energy events. This reduces the number of available events and causes large statistical errors in the first method. The second method compares the observed π^0

rate with the predicted rate. Unfortunately, there is a large uncertainty in the cross section of π^0 events. This problem can be somewhat mitigated by comparing the π^0 rate with rates of other two ring events in which pion production is involved, as proposed in [39]. After the analysis presented here was essentially complete, we have published a result that disfavors the ν_s hypothesis [40].

In order to utilize π^0 events, it is crucial to correctly identify them. When this work started, almost all of the Super-Kamiokande analysis was based on single ring events. Multiple ring events are harder to reconstruct correctly. As noted above, π^0 events are two-ring events. They can be misidentified as single ring electron-like events due to missing of the second ring (when it is weak), or due to close overlapping of the two rings. It is important to have a good $\pi^0 - e$ separation. A specialized π^0 fitter was developed, described in detail in this work. It was applied for extraction of neutral current events by selecting π^0 events that were subsequently used to analyze the neutrino oscillation and to discriminate between $\nu_\mu \leftrightarrow \nu_\tau$ and $\nu_\mu \leftrightarrow \nu_s$ cases. Along the way, a new event display program was developed with many flexible options, with the ability do show quantities used internally by fitters (for example, likelihood, or various expected light distributions), that is described in detail here.

The results of neutrino oscillation experiments open a window on new physics beyond the Standard Model. A new generation of experiments is coming up to explore the new area. One is MINOS at the NuMI beam line [36], that has started operation in March 2005. The NOνA experiment has been proposed to use the NuMI off-axis beam to search for electron neutrino appearance [37]. Another is T2K (Tokai to Kamioka), a 295 km baseline off-axis neutrino oscillation experiment that produces ~ 0.8 GeV ν_μ beam [38]. One of the goals is to discover ν_e appearance in $\nu_\mu \leftrightarrow \nu_e$ oscillation. A major background for ν_e events are π^0 events. A specialized fitter for $\nu_e - \pi^0$ discrimination, based on the one described here, will help reduce the background. T2K also plans to search for $\nu_\mu \leftrightarrow \nu_s$ oscillations as a fraction of $\nu_\mu \leftrightarrow \nu_e/\nu_\tau$ oscillations.

The rest of this document is organized as follows:

- Chapter 2 describes the Super-Kamiokande detector, including the principle of operation and the various components.
- Chapter 3 describes the the calibration methodology.
- Chapter 4 explains how the interesting events are selected from the vast background, using multi step data reduction.
- Chapter 5 describes reconstruction algorithms: vertex selection, ring finding, particle type identification, and momentum measurement.
- Chapter 6 describes the simulation of atmospheric neutrino production, interaction with water molecules, propagation of secondary particles, propagation of Cherenkov light in the detector, and response of active instruments in the detector.

- Chapter 7 introduces the event display, “tscan”, that was used to develop subsequent steps of the analysis, including its features and internals. An unexpected benefit of this event display was its role in science popularization.

- Chapter 8 describes in detail the developed multi ring fitter, that was used for the analysis. It also describes modular algorithms that are used by the fitter: expected light generation, including scattered light and normalization, working with “patches” of the detector, “charge inertia tensor” calculation, likelihood calculation, and simplex fitting. This chapter also describes fh2h—a Fortran to C header translator.

- Chapter 9 presents selected atmospheric neutrino sample and explains how various two ring samples were selected, using the developed tools.

- Chapter 10 presents the analysis using the two ring samples, including π^0 analysis.

- Chapter 11 summarizes this work.

Chapter 2

Detector

2.1 Cherenkov Radiation

Cherenkov radiation [41] is emitted by charged particles moving through a transparent medium with a speed v greater than the speed of light in that medium, where

$$v = \beta c > c/n \ , \tag{2.1}$$

that is, $\beta > 1/n$, where n is index of refraction, and β is the particle speed measured in units of the speed of light c.

The charged particles polarize the molecules, which then return back rapidly to their ground state, emitting prompt radiation. The emitted light forms a coherent wavefront if $\beta > 1/n$. Cherenkov light is emitted under a constant Cherenkov angle with respect to the particle track, in a forward cone (Fig. 2.1) with half opening angle

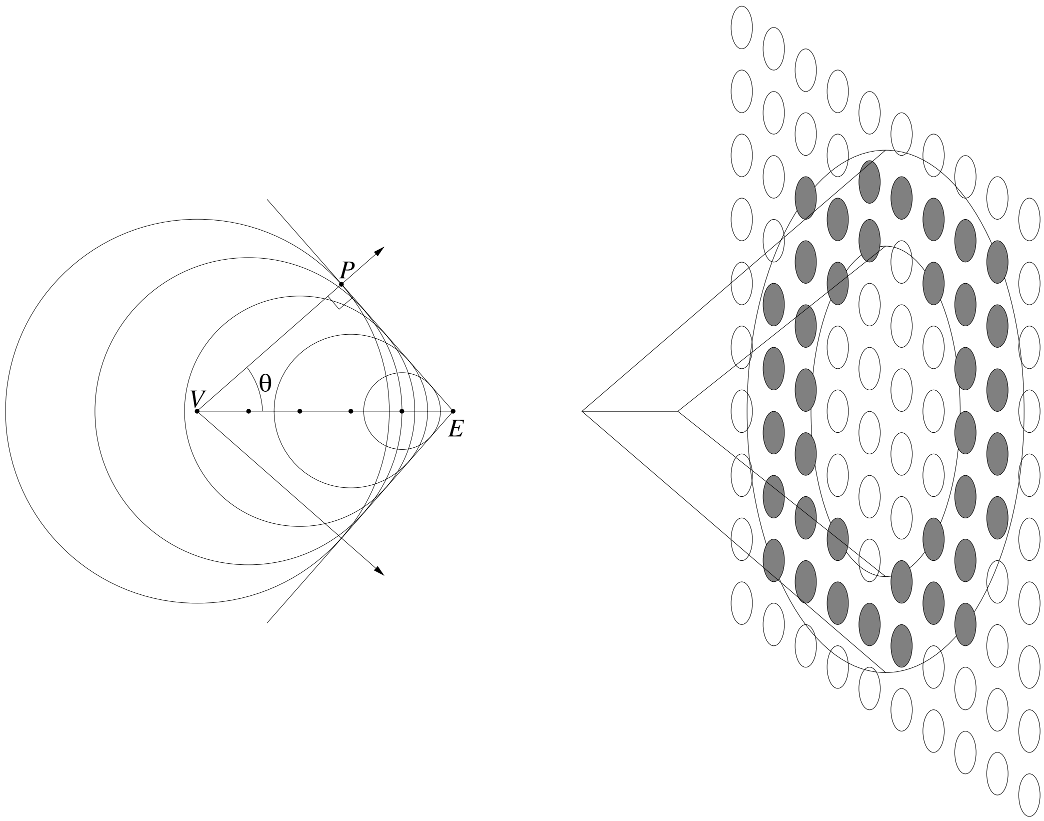


Figure 2.1: Schematic drawing of Cherenkov light and ring pattern generation. Left: light front produced by a particle moving with a speed greater than the speed of light propagates at an angle θ with respect to the track direction. It produces a Cherenkov light cone. Right: when the Cherenkov light arrives at the detector wall, it produces a ring pattern.

of

$$\cos\theta = \frac{\overline{VP}}{\overline{VE}} = \frac{\frac{1}{n}ct}{\beta ct} = \frac{1}{n\beta} \; . \tag{2.2}$$

For highly relativistic particles, such as what is most commonly seen in Super-Kamiokande, the particle travels with a speed that is close to the speed of light $v \simeq c$, i.e., $\beta \simeq 1$, and the opening angle is maximal, is given by

$$\cos\theta = 1/n \; . \tag{2.3}$$

For water, with an index of refraction of $n = 1.33$, the Cherenkov angle is about

42 degrees. For slower particles, the angle is smaller and the particle produces a so called "collapsed cone". When the light cone hits the detector wall, it produces a ring pattern.

The number of photons produced per unit path length of a particle with charge ze and per unit energy interval of the photons is

$$\frac{d^2N}{dEdx} = \frac{\alpha z^2}{\hbar c} \sin^2 \theta \, , \tag{2.4}$$

or, equivalently,

$$\frac{d^2N}{dxd\lambda} = \frac{2\pi\alpha z^2}{\lambda^2} \left(1 - \frac{1}{\beta^2 n^2(\lambda)} \right) \, . \tag{2.5}$$

The spectrum of Cherenkov light in water, after taking into account the transmission of water, ranges from 300 to 600 nm. It peaks at 430 nm. When seen with a naked eye (for example, when looking at uranium rods submerged in water in a nuclear reactor), it has a bluish hue.

2.2 Super-Kamiokande Detector

Super-Kamiokande [42] is a 50 kiloton water Cherenkov detector located underground in the Mozumi zinc mine, of the Kamioka Mining and Smelting company, near Kamioka in Gifu prefecture, Japan. A 2 km long horizontal tunnel, wide enough for large trucks, leads from the entrance at a side of Mt. Ikenoyama to the detector site. A secondary access is available through another rail tunnel. One km of rock overhead constitutes 2700 meters-water-equivalent of shielding from the cosmic rays

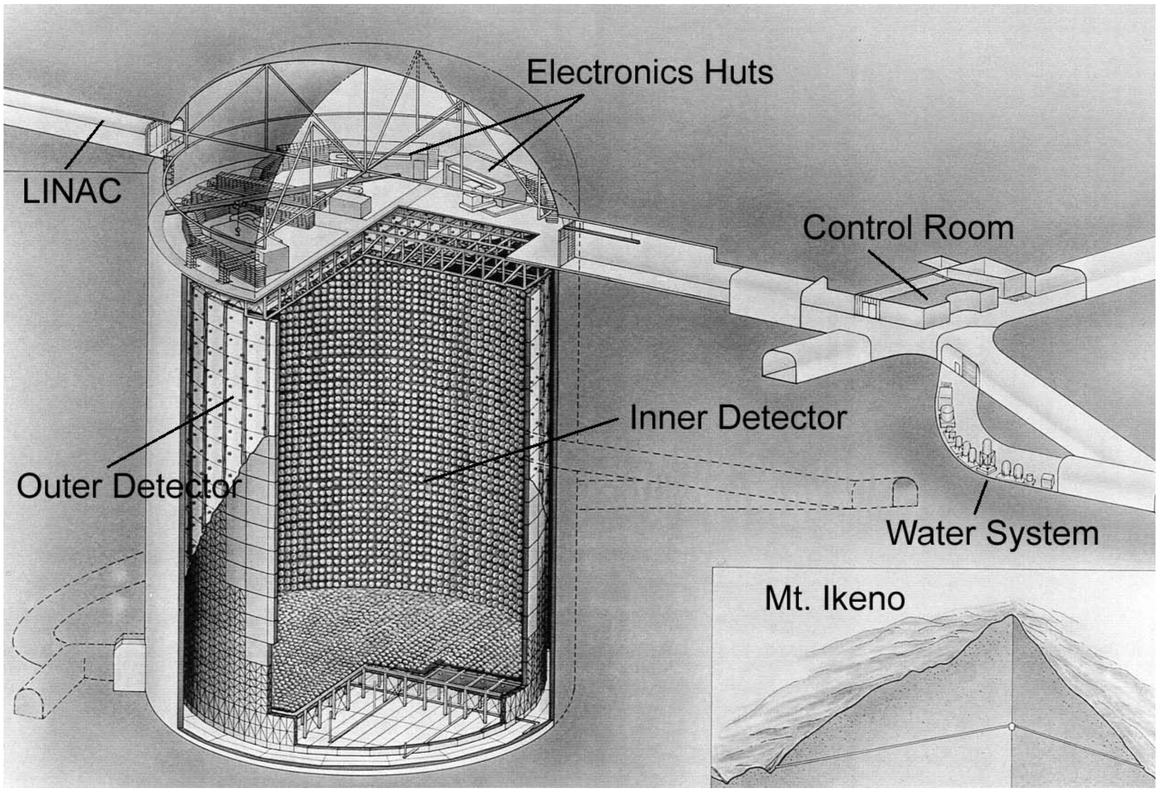


Figure 2.2: Schematic view of the Super-Kamiokande detector.

and reduces the muon rate to 3 Hz.

The detector (Fig. 2.2) is housed in a large cylindrical cavity with a hemispherical dome ceiling. The ceiling is lined with a coating called Mineguard (a spray-applied polyurethane membrane) in order to keep radon out of the area. Radon constitutes a significant background in low energy analysis (solar neutrinos), but it is not important for the high energy analysis (including atmospheric neutrinos).

Water is held in a welded stainless steel tank 39 m in diameter and 42 m high. Access to the tank is through a set of two hatches on the tank top, and a hatch providing access to the tank bottom. The top wall of the stainless steel tank also comprises a floor that supports electronics huts and other equipment and on which people can walk. Between the tank wall and the rock there is a layer of steel reinforced concrete.

The detector volume is divided optically into two major regions. The main, inner volume is cylindrical in shape and has diameter of 33.8 m and height of 36.2 m. It is viewed by 11,146 50 cm photomultiplier tubes (PMTs) [43]—7,650 on the side walls, 1,748 on the top, and 1,748 on the bottom. The PMTs are placed on a regular square grid with 71 cm step. The photocathode covers 40% of the inner detector (ID) surface. The space between PMTs is lined with a thin opaque black polyethylene telephthalate sheet. It constitutes one of two layers of optical division. The black sheet is installed at the level of the PMTs "equator", i.e., their widest place.

Due to the large size of the PMTs, the electron travels a long way in vacuum from the photocathode to the first dynode. The PMTs are thus very sensitive to a magnetic field. Even the Earth's magnetic field (450 mG in the mine) would systematically bias photoelectron trajectories and timing in the PMT. We reduce the field to below 50 mG with 26 sets of horizontal and vertical compensating Helmholtz coils which surround the detector at the stainless steel wall (inside the tank).

The inner detector is completely surrounded by an outer detector (OD). It has a shape of a cylindrical shell, 2.05 m thick at the sides and 2.2 m thick at the top and bottom. The OD is instrumented with 1,885 outward-facing 20 cm PMTs (302 on the top layer, 308 on the bottom, and 1,275 on the side wall) with acrylic wavelength shifter plates attached to them in order to improve light collection. The outer wall of the OD is the stainless steel tank which is covered with a white DuPont Tyvek material. The reflective surface improves light collection. The inner wall is covered with Tyvek which is white on the side facing outside and black on the inside facing side. This sheet is the second light barrier which separates ID and OD. Behind every OD PMT there is a round Tyvek collar which improves light tightness. The outer detector volume serves as an active veto counter against incoming particles, as well as passive shield for neutrons and γ rays from the surrounding rocks.

Between the inner and the outer detectors there is a 55 cm wide dead space that contains stainless steel beams which provide the PMT support structure.

2.3 Water System

Water in Super-Kamiokande has to be very clean in order to achieve high transparency (long attenuation length) and low radioactivity in order to reduce background for low energy events. Figure 2.3 shows a schematic view of the water purification system.

The source of water is a stream with spring water flowing in the mine tunnel. After the tank is filled through the water purification system, during normal operation, water is recirculated in the tank at rate of 360 ton/hour.

Water first passes through a 1 μm filter which removes small particles which reduce water transparency and provide a possible radon source. The heat exchanger cools

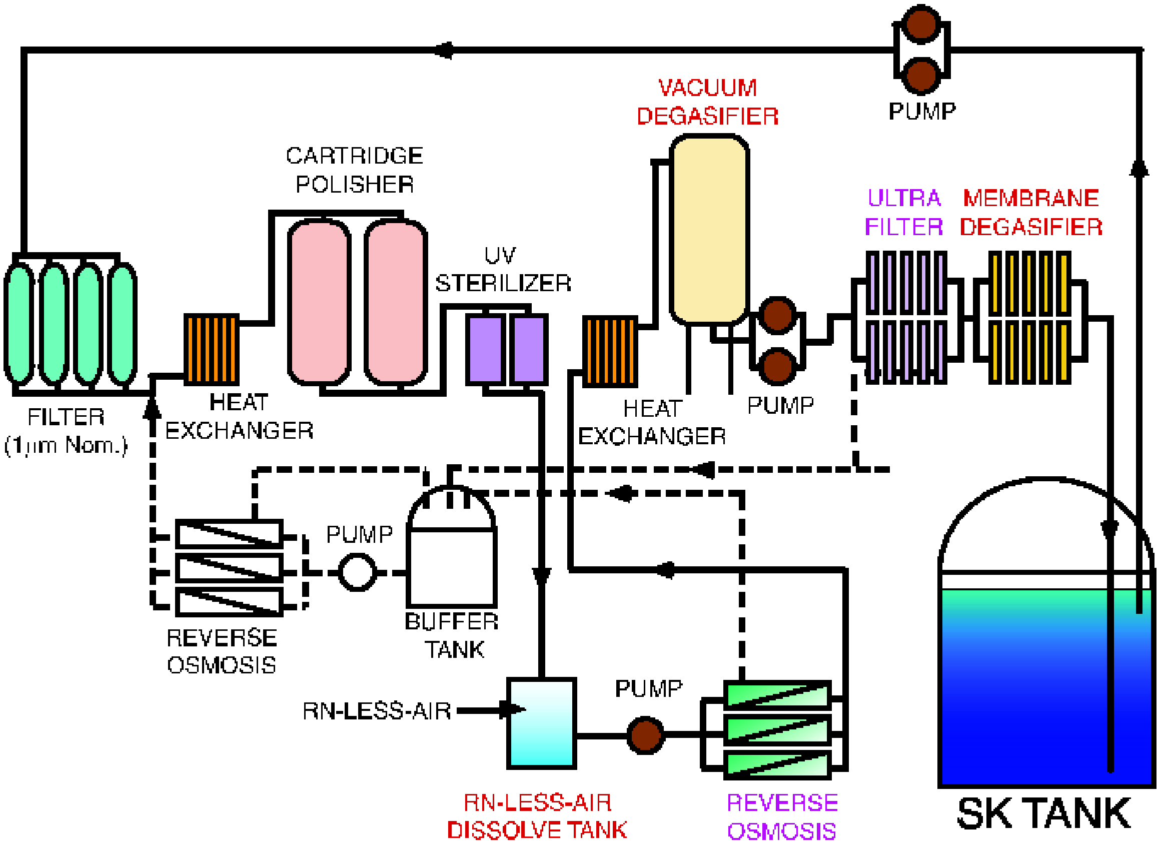


Figure 2.3: Schematic view of the Super-Kamiokande water purification system.

water to 13 degrees in order to reduce PMT dark noise and slow down growth of bacteria and other microorganisms. The ultraviolet sterilizer kills any remaining microorganisms. A cartridge polisher removes metal ions which reduce water transparency and contribute to radioactivity. Resistivity of circulating water is increased from 11 MΩ·cm to 18.24 MΩ·cm, which approaches the chemical limit. Radon-reduced air is dissolved in order to increase radon removal efficiency in the later vacuum degasifier stage. A reverse osmosis unit filters out molecular size impurities.

The vacuum degasifier removes dissolved air, radon and other gases. It consists

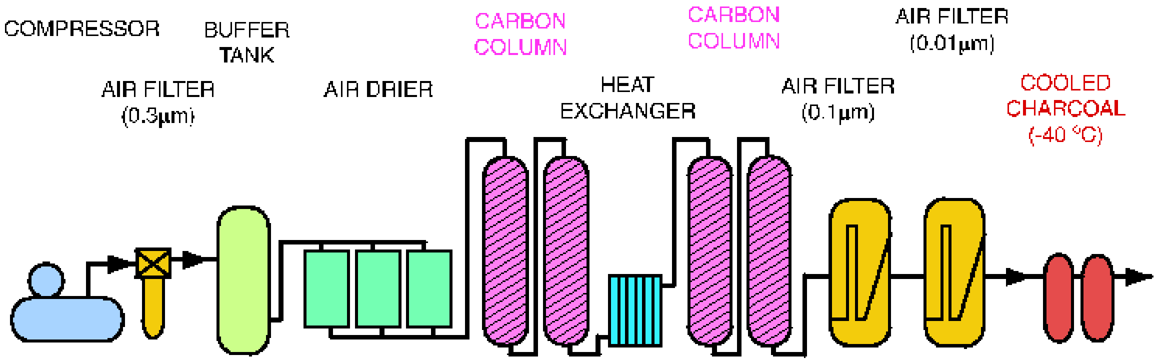


Figure 2.4: Schematic view of the Super-Kamiokande radon reduction system.

of a cylindrical 4.5 m tall stainless steel vessel in which water cascades down while a vacuum pump removes gas. Radon is a source of radioactivity, and oxygen encourages growth of many kinds of microorganisms. Radon removal efficiency is about 96%. Concentration of dissolved oxygen is reduced to 0.06 mg/l.

The ultrafilter removes particles down to 10 nm in size (about 10,000 molecular weight). It consists of hollow membrane filters. It reduces number of particles of size greater than 0.2 μm from 1,000 to 6 in each cm^3.

A membrane degasifier removes radon from water. It consists of thirty hollow fiber membrane modules and a vacuum pump. A radon reduced air is supplied as purge gas. Radon is removed with an efficiency of about 83%.

2.4 Radon Reduction System

Radon levels in the mine in the vicinity of the Super-Kamiokande detector can reach thousands of Bq/m^3. In order to keep radon level in the dome area and in the water system below 100 Bq/m^3, a radon reduction system was constructed. The radon reduction (air purification) system is located in the so called “Radon Hut” near the Atotsu tunnel entrance. A schematic view of the radon reduction system is shown on Figure 2.4.

There is a seasonal variation of radon levels in the mine tunnel air. In the area near the tank cavity dome, radon levels typically reach 2,000–3,000 Bq/m^3 during the warm season, from May until October, while from November to April the radon level is approximately 100–300 Bq/m^3. This annual variation is apparently due to convective air flow pattern in the the mine tunnel system. During the cool season, air in the mine is warmer and less dense than the air outside, and fresh air is sucked up into the Atotsu tunnel entrance and has a relatively short path through exposed rock before reaching the experimental area. During the summer, air outside is warmer than the air in the mine, and dense mine air flows down and out of the tunnel, drawing radon-rich air from deep within the mine past the experimental area.

The radon reduction system consists of of three compressors, a buffer tank, driers, filters, and activated charcoal filters. Fresh air from intake (initially located at the Radon Hut), is continuously fed into 40 hp air blower, with 10 m^3/min at 15 PSI pump capacity. The air is then pumped through filter, dehumidifier, and carbon filter tanks. A total of 8 m^3 of activated charcoal is used. The last 50 l of charcoal is cooled to -40 degrees to increase removal efficiency of radon.

The air is finally pushed through a smooth PVC air duct along the 1.8 km Atotsu tunnel to the experimental area at the rate of 10 m^3/min.

Originally, a second blower and air heater were also operated to regenerate radon-saturated carbon in the filter tanks at regular intervals, and lower the radon level of the output air down to 10 Bq/m^3 or less. However, maintenance and expert supervision of the carbon regeneration turned out to be impractical for continuous running. An extended intake air pipe was installed at a location approximately 25 m above the Atotsu tunnel entrance, where radon level concentration was found to remain at 10–30 Bq/m^3 all year long. It was decided that this low level satisfies the air quality goals and that carbon filter regeneration operations would no longer be required. The air flow from the radon hut keeps the radon levels in the experimental area at approximately 30–50 Bq/m^3 throughout the year.

2.4.1 Radon Reduction System Control

All the equipment, including motors, switches, indicator lights, valves, temperature sensors, pressure sensors, carbon monoxide sensors, radon sensors, etc. are automatically operated using a GE Fanuc PLC system. The system is programmed in ECLiPS state logic language. The software environment consists exclusively of many independent state machines—it is not a procedural language. Each state machine can transition to a new state based on several criteria, including a signal from another state machine, a sensor value reaching or leaving a predefined range, or after a preset amount of time passes. The control program is designed to react safely in case of any unexpected condition, putting components in the right order and timing into a safe state (for example, a blower can't blow against a closed valve, and it takes some time for a blower to stop).

A software development kit (SDK) on a PC uses a serial link to upload the compiled program to the PLC and to perform debugging. The same link can be used to interact with the program to check status and to control the state of the radon system. As there is no network connectivity in the radon hut, a second serial link connects the system with a Sparc station in the control room 2 km away. As RS-232C serial connection can't work at such large distance, a simple modem on a dedicated cable is used. This gives the shift personnel the capability to remotely control and check the radon system.

2.5 Inner Detector

The inner detector PMTs are custom made Hamamatsu R3600 tubes [43]. The bialkali (Sb-K-Cs) photocathode has peak quantum efficiency of about 21% at wavelengths of 360 nm–400 nm. The collection efficiency for photoelectrons at the first dynode is over 70%. The PMTs were specially designed to have good single photoelectron (pe) response (timing resolution). The transit time spread for a 1 photoelectron (pe) signal is 2.2 ns. The average dark noise rate at the 0.25 pe threshold is about 3 kHz. The ID PMTs were operated with gain of 10^7 at a supply high voltage 1,700–2,000 V.

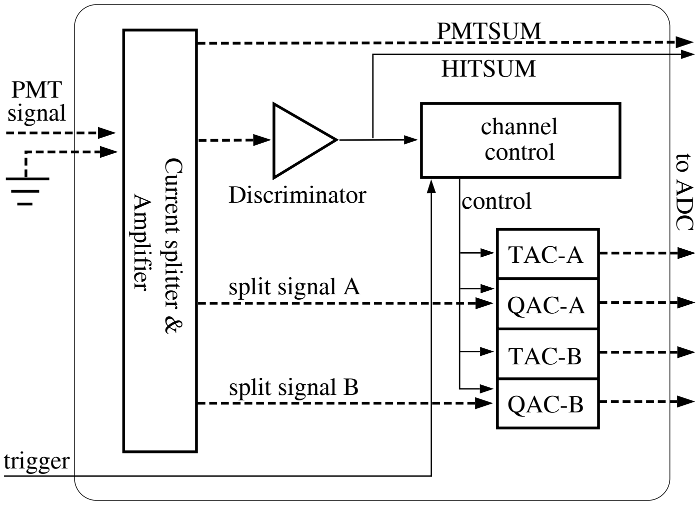


Figure 2.5: A schematic view of one channel of the analog input blocks of the ATM. Dashed arrows show the PMT signal as it is processed. Solid arrows show logic signals.

The cables carrying high voltage and signal exit the PMTs through water-tight fittings and run up along the surface of the outer detector region. They exit the Super-Kamiokande tank through 12 portholes which are designed to block light and radon. The cables are of equal length so that the signal travels for the same amount of time from each PMT. This helps in forming a coincidence trigger. The extra cable length from PMTs close to the surface is laid out in large cable trays on the surface of the tank.

The cables enter 4 counting houses (huts) positioned at the perimeter of the tank top. Each hut supplies high voltage and collects signals from a quarter of the tank. There is a fifth central hut which, among many other things, collects signals from the four peripheral huts.

The cables enter custom built Analog Timing Modules (ATMs) [45] housed in

TKO (Tristan KEK Online) [46] crates. The ATMs analyze the raw PMT signal, extract time and charge information, and perform the first step in trigger formation.

The ATM has the functionality of a combined ADC (Analog to Digital Converter), and TDC (Time to Digital Converter), and records the integrated charge and arrival time of each PMT signal. One ATM board processes signals from 12 PMTs. Figure 2.5 shows schematic view of the analog input block of the ATM. The PMT signal is fed into the current splitter and and is divided into four signals. One of them is fed to the discriminator with the threshold level for each channel set to -1 mV, which corresponds to 1/4 pe equivalent. Each time the input signal to an ATM goes over the threshold the ATM outputs a square "hit" signal 200 ns long and 10 mV high. The hit signals of all PMTs are added in analog fashion in order to form a HITSUM signal that is used to form the global trigger. When the HITSUM signal exceeds a certain level, a trigger is issued. In other words, a trigger is issued when a certain number of PMTs fire within a 200 ns timing window. At the same time, either of the split signals, A or B, is held by the QAC (Charge to Analog Converter), and a TAC (Time to Analog Converter) starts to integrate a constant current [47]. If a global trigger is received, the information in the TAC/QAC is digitized and stored in internal memory buffers. Since, for each channel, TAC integration is started by the PMT signal itself, the arrival time can be inferred from the value of the integrated charge. A "channel control" chip generates start/stop signals for the TAC, the gate signal for the QAC, and clear signal for both. The ADC/TDC output data have

12 bit resolution. Charge is digitized with a resolution of 0.2 pC and a range of about 450 pC. One photoelectron (pe) is 2.055 pC. Time information has a resolution of 0.4 ns with a range of about 1,300 ns.

There are three kinds of ID hitsum triggers: high energy (HE), low energy (LE), and super low energy (SLE), each set at a different level. For example, typically, the HE trigger is set at -340 mV, LE at -320 mV and SLE is adjusted lower and lower as the equipment is upgraded to cope with higher and higher trigger rates. The LE level corresponds to 29 hits.

In order to reduce dead time, there are two sets of integrating circuits (two QACs and two TACs) for each PMT. This way, two successive signals, for example, a muon and an electron from its decay, can be recorded with high efficiency.

The ATMs are temperature sensitive. Because of this, the temperature inside of each hut is kept within 1 degree by means of a large air conditioning system. Also, during data taking, pedestal information is taken automatically every 30 minutes.

High voltage for each ID PMT is distributed by 48 CAEN SY527 high voltage main frames. Each SY527 main frame accommodates 10 CAEN A933K high voltage cards which have one primary high voltage generator and 24 programmable active distributor output channels on each card. These main frames are connected to a CAENet network via four sets of V288 CAENet VME interface cards to be controlled and monitored remotely by a slow control computer.

Figure 2.6 shows a schematic view of the inner detector data acquisition system.

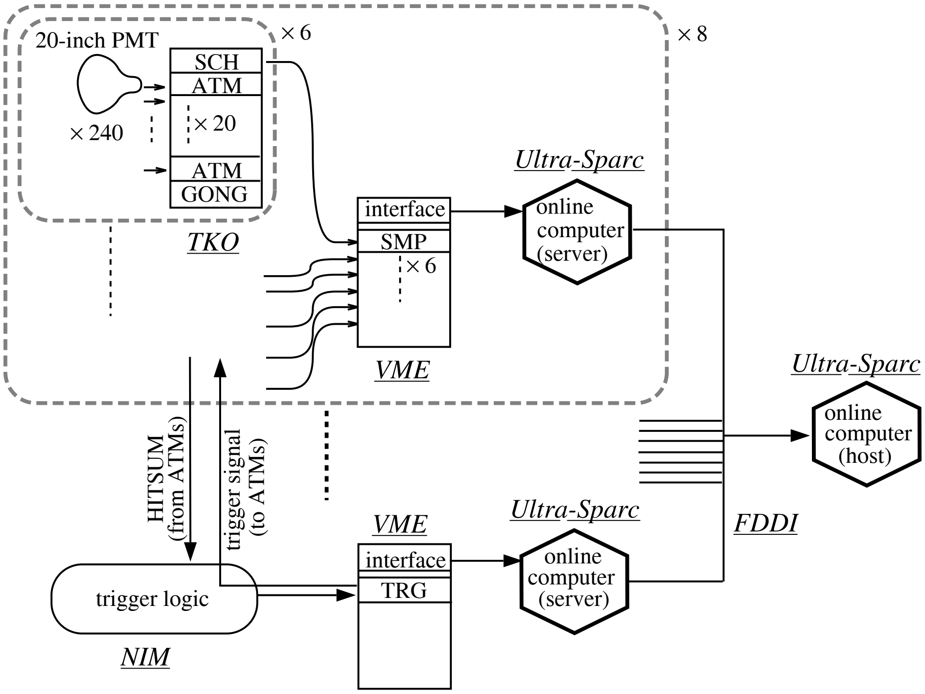


Figure 2.6: Schematic view of ID DAQ data flow.

946 ATM modules are located in four quadrant huts. ATMs record ADC/TDC data of each above threshold PMT when a global trigger threshold is asserted by the VME TRG module. The global trigger signal, and event number information generated in the TRG module, are distributed to all ATMs via GONG (Go/NoGo) modules. Digitized data stored ATM FIFO memory is stored into "Super Memory Partner" (SMP) VME memory modules.

Each online server (200 MHz UltraSparc CPUs, `sukon1`–`sukon8`, two per hut) reads event data from SMP memory via a Bit-3 Sbus-VMEbus interface using block transfer. Trigger information, i.e., the event number, the trigger time, and trigger type are recorded in VME TRG module, which is read by a separate server.

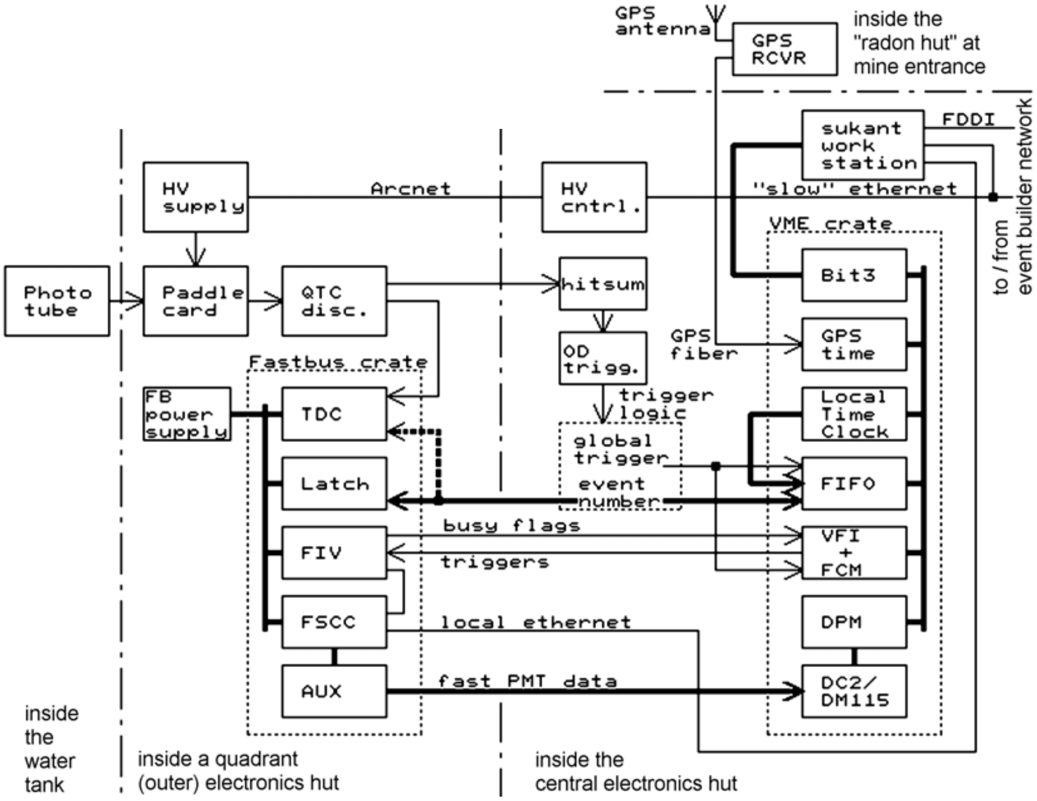


Figure 2.7: Block diagram the Outer Detector data acquisition system.

All data collected by the servers is transferred to the online host computer `sukonh` (a 300 MHz UltraSparc CPU), located in the control room, via an FDDI network, and the data (including OD data) is merged by an event builder process to assemble a complete event record.

2.6 Outer Detector

The outer detector PMTs are Hamamatsu R1408 tubes, most of which were salvaged from the IMB experiment.

To enhance light collection, each PMT is equipped with a $60 \times 60 \times 1.3$ cm wavelength shifter plate. The plates are made of acrylic, doped with 50 mg/l of bis-MSB which absorbs Cherenkov light and emits light at wavelength to which PMTs are more sensitive [44]. The plates improve light collection efficiency by 60%. The price paid is degradation of the timing resolution from 13 to 15 ns, which is acceptable for the functioning of the OD.

A single coaxial cable carries both high voltage and signal for each tube. The cables run along the inner detector cables. The cables are all of equal length, except for several cables from PMTs at the bottom of the detector that had to be extended.

The cables connect to custom paddle cards located in two crates per hut. The paddle cards separate AC PMT signals from DC high voltage. High voltage is provided by four LeCroy 1454 High Voltage Mainframes, one per quadrant hut. Each HV mainframe controls 48 channels of high voltage, although only 40 channels are actually used. Each HV channel feeds one paddle card to which 12 PMTs are connected. OD PMTs are operated at about 1,800 V. High voltage of individual PMTs cannot be adjusted. Because of this, PMTs are matched by gain. Each PMT can be disconnected by pulling a metal jumper from a paddle card. Most often PMTs are disconnected when they draw too high a current or when they are identified as flashers. The paddle cards pick off analog PMT signals through a high voltage capacitor and resistor network, allowing a single cable to be used for both high voltage and signal.

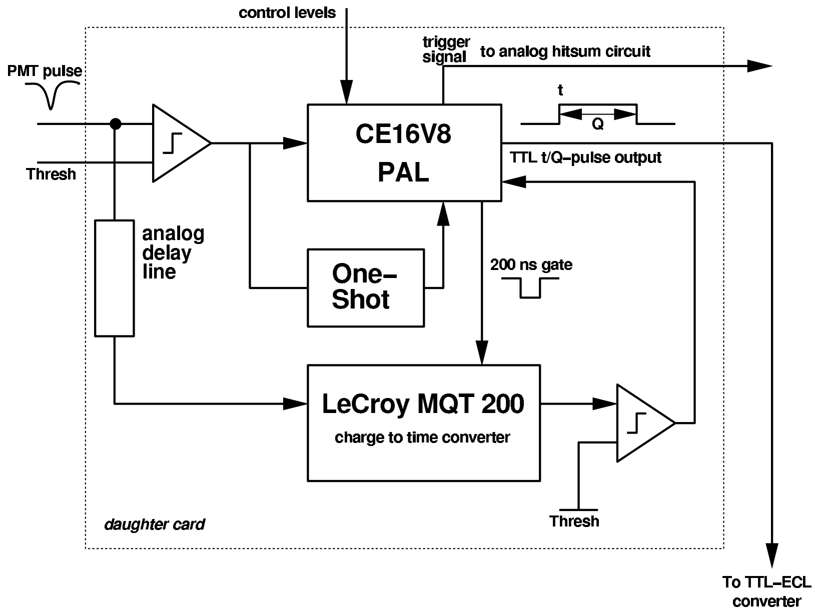


Figure 2.8: Block diagram of a QTC module.

Signals from the paddle cards are carried by coaxial ribbon cables to custom built Charge to Time Converters (QTCs—see Fig. 2.8). The QTC analyzes incoming PMT signals and outputs a square signal (ECL level logic pulse) which encodes time and charge information. Time is encoded in the timing of the leading edge, charge is encoded in the width of the pulse.

The QTC also generates a hitsum signal for formation of the OD trigger. Every time the signal from a PMT goes over a threshold of −25 mV, which corresponds to about 1/4 pe, a square pulse 20 mV high and 200 ns long is produced. The OD hitsum signals are added up similarly to the ID hitsum. The OD trigger level is set at a level of 19 hits.

Figure 2.7 shows a block diagram of the OD data acquisition system. The ECL pulses from the QTCs travel through ribbon cables to multi-hit LeCroy 1877 Pipeline Time to Digital Converters (TDCs) in Fastbus crates, where times for all leading and trailing edges are recorded. Each TDC module has 96 input channels (1 per PMT), with five TDC modules per Fastbus crate and one Fastbus crate in each electronics hut, providing total capability for 1,920 TDC channels.

Each TDC channel works as a circular buffer that can store times of up to 8 QTC pulses (16 edges with 2 edges, leading and trailing, per PMT hit) with a resolution of 0.5 ns over a window of up to 32 μs width. If more than 8 PMT pulses occur during a TDC window, then only the last 8 pulses are recorded and any earlier ones are lost. When the trigger arrives, the TDC digitizes the times of the last 16 edges, Only edges within 10 μs before and 6 μs after the global detector trigger are digitized. This time window used to be wider, but was narrowed in order to avoid losing the earliest hit due to multiple after-pulsing signals.

All TDC channels are digitized and read out simultaneously following a fixed delay (6 μs) after a global detector trigger. During digitization, which can take 2 to 15 μs depending on the number of edges in the event, the OD electronics cannot record new data (busy/in-progress dead time), but after read-out the channels are free to take new data. If a global trigger signal is received during OD dead time, a flag bit is introduced into the data stream to record this condition.

The digitized hits are read by a Fastbus Smart Crate Controller (FSCC) which sends the data to the central hut via a 32-bit RS485 cable. In the central hut, the data passes through a Motorola DC2/DM115 slave CPU in the OD-DAQ VME crate and is stored in a large VME dual-port memory buffer (DPM) The data is read from DPM via a Bit3 VME-SBus interface by an online computer, a Sparc20 workstation `sukant`. `Sukant` assembles the data and sends it to an event builder on `sukonh` which builds the complete event from ID and OD data.

2.7 Online Data Processing

Data from event builder on `sukonh` is sent to three destinations. The main data stream is sent to a reformatter process on the `sukrfm` computer. This process transforms the data into ZBS format, based on the ZEBRA library developed at CERN. The second stream is sent to the `kingfish` computer which runs the online event display. A third stream is sent to the intelligent trigger computers for filtering the SLE events.

The reformatter sends data to the `tqreal` process running on the `sukop` computer. `tqreal` translates raw ADC and TDC counts to charge and time using calibration data. The calibrated data is then sent to a robotic magnetic tape library.

The amount of data processed is 450 kbytes/s, corresponding to 40 Gb/day or 17 Tb/year.

Chapter 3

Calibration

Basic detector calibration such as PMT charge and timing response and water transparency measurements are performed regularly.

3.1 Relative Gain

The high voltage of each PMT is adjusted in order to get a uniform gain response throughout the detector. Relative gain is measured using the setup shown in Figure 3.1. Light from the Xenon lamp goes through a filter which passes only ultraviolet radiation. Intensity can be adjusted by a neutral density (ND) filter. The light enters an optical fiber which is lowered into the Super-Kamiokande tank. At the end of the fiber is a scintillating ball. The spherical ball is made of acrylic doped with a BBOT (2,5-bis(5'-tert-butyl-2-benzoxazolyl)thiophene) scintillator which absorbs UV light and emits light close to the Cherenkov spectrum in water, with a peak

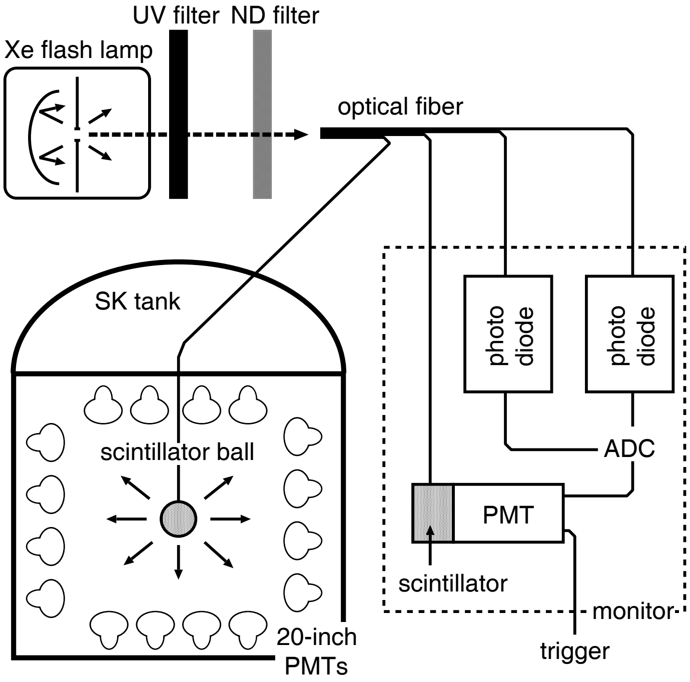


Figure 3.1: Schematic drawing of the scintillating ball calibration system.

at 440 nm. The ball also contains MgO, a diffuser powder which makes the light emitted from the ball isotropic. Each PMT detects a few tens of pe.

The input light intensity is monitored and an SK trigger is formed with a two inch PMT and two photodiodes. The output of the monitor PMT is also used for triggering. The light intensity measured with 20 inch PMTs is corrected for attenuation, distance, angular acceptance, and the light intensity measured by the monitoring system. The relative gain G_i of the i-th PMT is obtained by:

$$G_i = \frac{Q_i}{Q_0 f(\theta)}\, l_i\, e^{l_i/L_{\rm att}} \,, \tag{3.1}$$

where Q_i is the observed charge by the PMT, Q_0 is a constant, $L_{\rm att}$ is the attenuation length, l_i is the distance from the scintillator ball to the PMT, and $f(\theta)$ is the PMT acceptance function that depends in the incident angle of the light on the PMT.

The corrected intensity is used to calculate high voltage adjustments. The high voltage value for each PMT is set so that the corrected charge of each PMT is approximately the same as for all others. Corrected charge is the pulse height corrected for light attenuation, acceptance, and uniformity of the scintillator ball. It is further normalized by the Xenon monitor pulse height. This measurement is done for various positions of the scintillator ball and settings of the high voltage value.

The remaining relative gain spread is 7 percent which is corrected for by the offline software.

3.2 Absolute Gain

The goal of the absolute gain calibration is to relate observed charge to number of photoelectrons. We used a nickel gamma-ray source for this purpose. At the source's center is radioactive Californium-252 which provides neutrons through a process of spontaneous fission. Approximately 97% of the time, ^{252}Cf decays via an α decay, while the remaining 3% of the decays undergo spontaneous fission. The half-life of Californium-252 is 2.56 years. An average of 3.76 neutrons are produced per fission. The average neutron energy is 2.1 MeV and the spectrum extends up to about 14 MeV. The source is surrounded by volume containing water tightly packed with nickel wire. The water acts as a moderator slowing down the neutrons. The neutrons lose energy by elastic scattering off the protons in water. It takes approximately 19 n-p scatterings (several cm of travel) and a few μs for a 2 MeV neutron to lose enough energy to be considered thermal. Thermalized neutrons are captured on nickel nuclei. Following the capture, gamma rays are emitted, having on average 6 to 9 MeV. They escape the source and produce an electromagnetic cascade in the Super-Kamiokande water. The energy is low enough that almost all of the Cherenkov light observed by PMTs are single photon hits. A clear single photoelectron peak was observed. From this distribution it was determined that on average a charge of 2.055 pC corresponds to one photoelectron.

3.3 Timing

The purpose of the timing calibration is the determination of the timing offset of each channel and the dependence of the offset on observed charge. The relative timing of PMT hits is important for event reconstruction. It depends on the length of the signal cable between the PMT and the ATM, and on observed charge because of discriminator slewing effect. The timing difference of each PMT has to be measured precisely to get better timing resolution.

Relative timing calibration is performed using a setup consisting of a laser, an optical fiber and a diffuser ball, shown in Figure 3.2. The nitrogen laser provides intense ultraviolet light of wavelength 337 nm. It has a short 3 ns pulse width and ± 0.5 ns time jitter. The UV light pumps a dye laser module which provides light with wavelength of 384 nm, close to the peak of the Cherenkov spectrum. The light

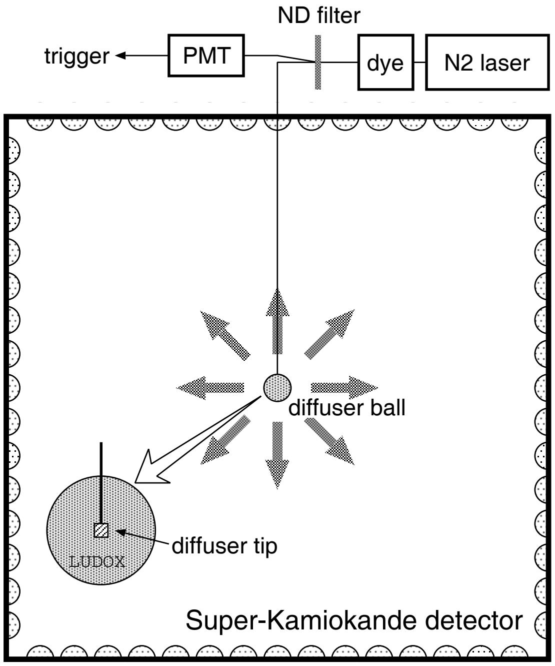


Figure 3.2: Schematic drawing of the timing calibration system.

passes through a variable attenuation filter which can modify the light intensity. The light is next guided to an optical fiber which is lowered into the Super-Kamiokande tank. At the end of the fiber, light enters a tip made of titanium dioxide suspended in optical cement. The light emitted from the tip is further diffused by LUDOX silica gel with 20 nm glass fragments. The combination of a diffuser tip and LUDOX can make a modestly diffused light without introducing significant timing spread. The diffused light is then registered by the detector's 20 inch PMTs. Part of the light is guided to a monitor PMT which generates a trigger signal. A TQ map is made for every channel. This timing dependence is corrected for in the analysis. The time resolution decreases (improves) with increasing charge. At the one photoelectron level the resolution is 2.8 ns RMS.

3.4 Water Transparency

Light traveling in water is attenuated through the processes of absorption and scattering. Intensity of light at distance l from the source is

$$I = I_0 \frac{1}{l^2} e^{-l/L_{\mathrm{att}}} , \tag{3.2}$$

where the attenuation length L_{att} is determined by

$$\frac{1}{L_{\mathrm{att}}} = \frac{1}{L_{\mathrm{abs}}} + \frac{1}{L_{\mathrm{scat}}} . \tag{3.3}$$

Two methods are used to measure water transparency: direct laser measurement, and through the observation of light from cosmic ray muons.

3.4.1 Laser Measurement

The attenuation length is measured directly by observing changes in brightness of a monochromatic light source of a controlled wavelength while changing the distance between the source and a detector. The setup is shown on Figure 3.3. A nitrogen laser with dye module produces a beam with tunable wavelength in the range of 337–580 nm. The light is guided to an optical fiber which has a diffuser ball attached at the other end. The ball is lowered into the Super-Kamiokande tank. A CCD camera located on the top of the tank views the ball. The intensity of the ball's image is

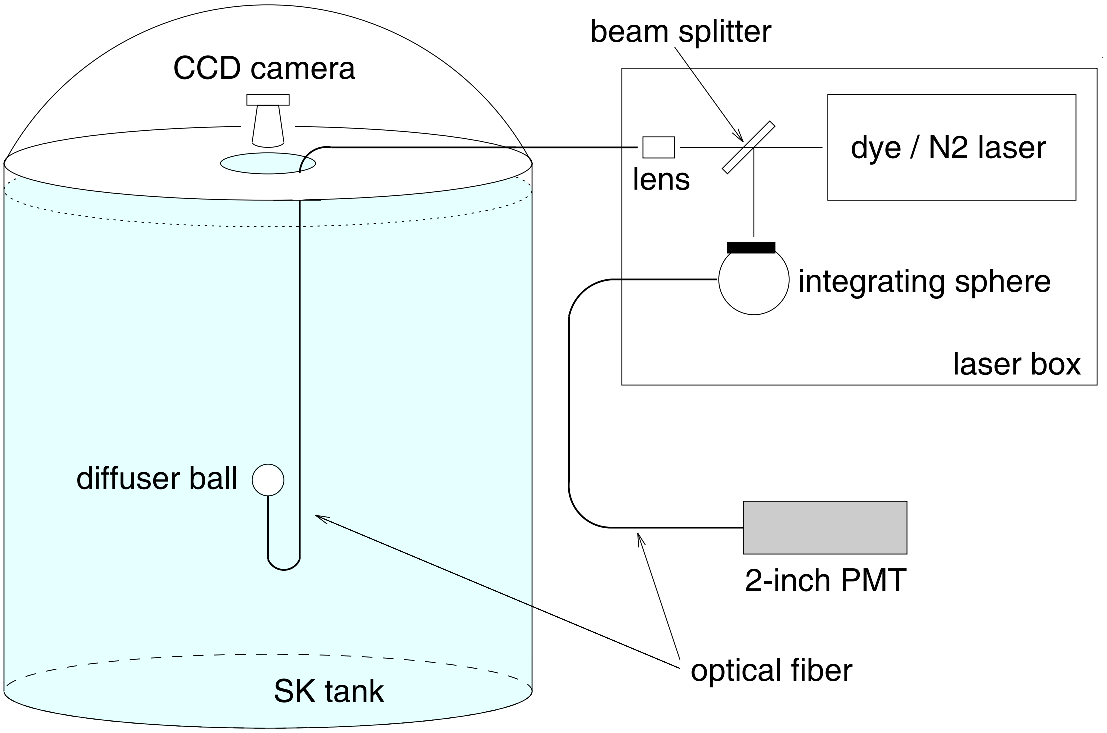


Figure 3.3: Schematic drawing of the laser water transparency calibration system.

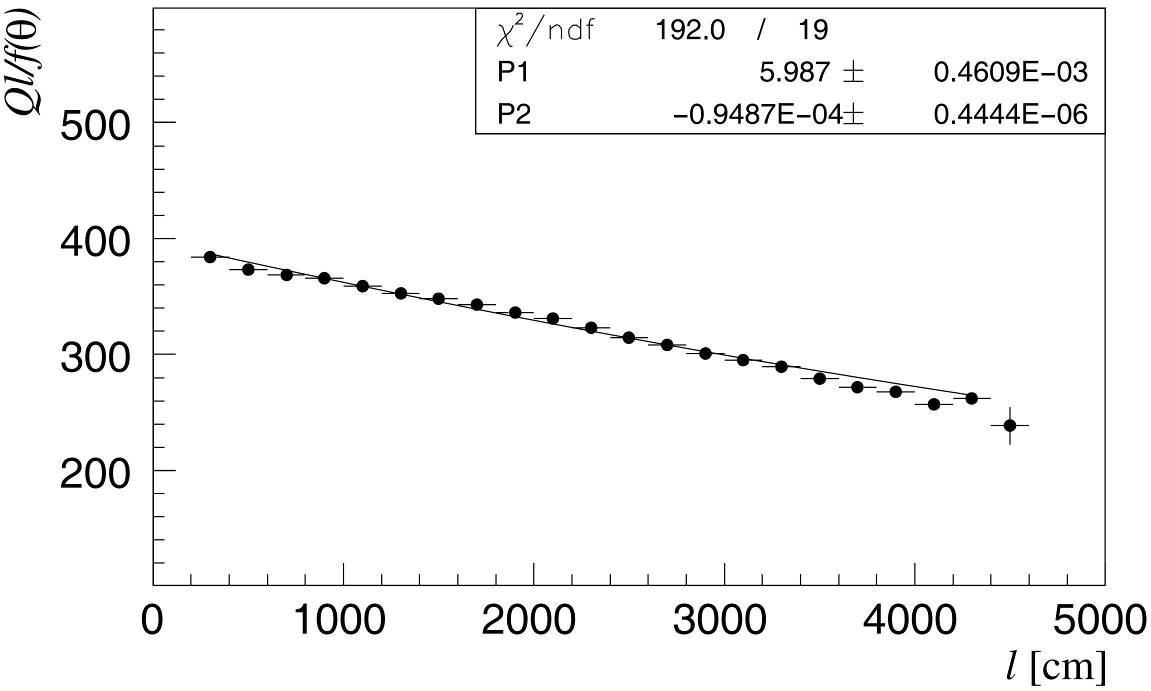


Figure 3.4: Effective observed charge from cosmic ray muons as a function of the light path length.

monitored as the depth of the ball is varied. Part of the laser light is split from the main beam for monitoring of laser intensity. The intensity of the ball's image is plotted as a function of distance and an exponential function is fit. The exponential "decay" constant is the attenuation length. The process is repeated for several light wavelengths.

3.4.2 Cosmic Ray Muons

Through going cosmic ray muons are also utilized for measurement of water transparency. They provide a measure of an effective attenuation length, averaged over the Cherenkov spectrum. These muons are energetic enough to deposit almost constant ionization energy per unit length (about 2 MeV/cm) independent of particle energy. This method has the advantage of providing a constant monitoring, without introducing any detector down time. In addition, it monitors PMT gain.

Charge produced by a direct light from the muon track reaching a PMT can be expressed as

$$Q = Q_0 \, f(\theta) \, \frac{1}{l} \, e^{-l/L_{\mathrm{att}}} \ , \tag{3.4}$$

where l is the light path length, L_{att} is the effective attenuation length, Q_0 is a constant, and $f(\theta)$ is the PMT acceptance function that depends in the incident angle of the light on the PMT.

Cosmic ray muons are selected which produce between 50,000 and 125,000 photoelectrons. The upper cut rejects energetic muons which produce additional showers. Muons which travel near the center of the detector are selected by requiring muon entrance and exit to be within a 15.9 m circle at the top and bottom of the detector. Approximately vertical muons are selected by requiring entry and exit points projected on the XY plane to be not further apart than 5 m. A constant number of photons per unit muon track length is assumed. Observed charge is corrected for distance and PMT acceptance. The distribution of corrected charge as a function of photon travel length is fitted with an exponential.

Figure 3.4 shows a typical result. It shows $Ql/f(\theta)$ as a function of the light path length l, together with best fit of the function defined in equation 3.4. The resulting attenuation length is found to be 105.4 ± 0.5 m.

Chapter 4

Data Reduction

The data is sent to the reduction processes for each analysis chain: low energy, upward-going muons and high energy.

High energy reduction starts by rejecting events with too few hits. This gets rid of high-rate, low-energy events.

The most critical and most powerful cuts in the data reduction, described below, involve the requirement of no outer detector activity.

Raw data is saved to tape at a rate of about 10^6 per day.

4.1 First Reduction

The first reduction step has several simple (fast to compute) cuts to eliminate most of the unwanted events. This step contains three cuts. The first cut slides a 300 ns wide timing window searching for the interval with most of inner detector hits. Events with less than 200 pe (photoelectrons) in the window are rejected. This threshold corresponds to electron momentum of 22 MeV/c and muon momentum of 190 MeV/c. This cut rejects low energy background, mostly radioactive decays.

A second cut checks the number of outer detector hits in a 800 ns wide fixed position timing window. Events with more than 50 hits are rejected. This cut rejects cosmic ray muons which enter the detector and make light in the outer detector.

A third cut rejects events which are preceded by an event less than 100 μs earlier. This cut rejects decay electrons from stopping muons, and fake events caused by reflected signals in the cables after a bright cosmic ray muon event and by "after-pulsing".

Also, any event which follows a fully contained candidate event within 30 μs is saved, as a possible candidate for a muon decay electron.

At this stage a very simple and fast energy estimate is made based on the total uncorrected charge observed by inner PMTs.

4.2 Second Reduction

The second reduction step consists of two cuts.

The first cut rejects events with more than 25 outer detector hits in an 800 ns window which have OD trigger bit set and for which total inner detector charge is less than 100,000 pe. This cut rejects more of the cosmic ray muon background.

The second cut rejects events for which most of inner detector charge in 300 ns window comes from one PMT. This rejects electrical noise and events extremely close to one PMT.

4.3 Third Reduction

At this stage, most of the background consists of noise, events and cosmic ray muons with few hits in the outer detector.

The first cut removes remaining cosmic ray muons. To be rejected, events have to satisfy three criteria. The first criterion requires the charge of the hottest PMT to be greater than 230 pe. PMTs at the muon exit point are typically saturated. The second criterion requires the goodness of the muon fit to be greater than 0.75. The fitter works by finding muon entry and exit points. Entry point is defined by earliest cluster of PMT hits. Exit point is found by selecting saturated PMTs. The third criterion requires there to be more than 9 hits in the outer detector in 800 ns timing window within 8 m of the reconstructed muon entrance or exit points.

The second cut removes entering stopping muons. Rejected events have an entrance point (cluster of earliest hits) found by the stopping muon fitter and a number of outer detector hits greater than 9 or greater than 4 with muon fitter goodness greater than 0.5.

The third cut removes entering stopping muons which enter the inner detector through one of four thick cable bundles on top of the tank and therefore produce very little or no light in the outer detector. This class of events was recognized after analysis of the first contained event sample. Since then, a veto detector was installed above each cable port. Each veto detector is a 2 m × 2.5 m square scintillator counter. Events with a reconstructed muon track crossing the horizontal plane of the veto counter within 4 m are rejected.

The fourth cut rejects flasher events. Many flashers have a long tail in the timing distribution. They are rejected by sliding a 100 ns timing window between 1200 and 1700 ns searching for a minimum number of hits. Events with more than 14 hits in the window or with more than 9 hits in the window and equal or less than total of 800 hits in the inner detector are rejected. Frequently, flashers create a characteristic pattern of light in the detector. If an event matches one of several characteristic patterns it is rejected.

The fifth cut rejects low energy events. These are mainly caused by radioactivity or low light flashers. Using the reconstructed vertex, the time of flight is subtracted from every PMT hit. Using this residual timing a sliding 50 ns wide timing window is applied and a maximum number of hits in the window is searched for. Events with less than 50 hits are rejected. This corresponds to visible energy of 9 MeV.

The sixth cut rejects accidental coincidence events. Sometimes a low energy event causes a trigger which is subsequently followed by a cosmic ray muon within the event timing window. Those events have not been rejected by previous cuts because there is little OD activity in the fixed timing window, and charge in the inner detector sliding window is large. Those events are rejected by looking at hits in the inner and

Reduction step	Data	Monte Carlo
Trigger	1,889,599,293	14,013.9 (100%)
First reduction	4,591,659	14,006.3 (99.95%)
Second reduction	301,791	14,006.1 (99.94%)
Third reduction	66,810	13,993.3 (99.85%)
Fourth reduction	26,937	13,898.1 (99.17%)
Fifth reduction	23,984	13,895.3 (99.15%)
Volume and energy cuts	12,180	13,676.7 (97.59%)

Table 4.1: Number of events after each reduction for fully contained sample during 1489 days of live time.

outer detectors in a fixed timing window between 1300 and 1800 ns. Events with more than 19 hits in the OD and more than 5000 pe in the ID are rejected.

4.4 Fourth Reduction

A more sophisticated flasher cut which compares charge patterns between events is now employed. The idea is that a flasher usually produces a distinct light light pattern which is repeated for every flash. This filter program divides the detector into 2 m × 2 m patches of 6 to 9 PMTs and remembers total light intensity in each of the patches for every event. It compares this intensity for every event in a given sample and rejects events which have a very similar pattern. Only 0.05% of atmospheric neutrino Monte Carlo events are rejected at this step.

4.5 Fifth Reduction

This reduction performs two cuts.

Events, which have more than nine outer detector hits in 200 ns time window, 100–8900 ns earlier than trigger time, are rejected. This cut eliminates invisible cosmic-ray muons whose energy is below Cherenkov threshold in the inner detector and triggered by its decay electron.

Cosmic ray muon hypothesis is fit to the event, and the same cut as the first cut in the third reduction is performed, using the more precise muon fit.

These events comprise the fully contained high energy sample. Subsequent analysis performs further selection, such as fiducial volume and energy cuts: vertex is required to be more than 2 m from the wall of the inner detector, and the visible energy is required to be greater than 30 MeV.

Chapter 5

Reconstruction

After the few interesting events are selected from the large number of background events, detailed reconstruction programs are run. These measure precise vertex position, find all the rings, and for each ring, measure direction and momentum, and determine particle type. Also, the number of electrons from muon decay is reconstructed.

5.1 Vertex Fit

The vertex is found initially by minimizing the width of the residual time distribution, i.e., time of flight subtracted PMT hit times. The point fitter algorithm works by assuming that all light is emitted from a single point—the vertex. It ignores any effects due to track length or scattered and reflected light. It takes into account the spectral dependence of the index of refraction of water.

The next step is finding the edge of the ring. Initial ring direction is estimated by summing all the charge as seen from the vertex. Only the main ring of a multi-ring event is considered. The opening angle and the direction of the ring are determined by requiring vanishing of the second derivative of charge as a function of opening angle, by maximizing the first derivative of charge as a function of the angle θ at the ring edge, by maximizing the amount of charge inside the ring, and by requiring the ring opening angle to be approximately equal to the nominal Cherenkov angle. PMTs at the ring edge are tagged.

The final step is a precise vertex fit. At this point scattered light and track length are taken into account. Sub-steps are executed in sequence iteratively.

Ring direction is refined by minimizing the difference between the angle θ to a PMT and an average angle θ of all tagged PMTs.

Next, non-zero track length is taken into account in order to calculate a more accurate residual time. For each PMT inside of the Cherenkov cone, a point along the track from which the light was emitted towards the PMT is calculated. In fact, two such points are calculated. First one is found by simple geometry projecting back the PMT onto the track at the Cherenkov angle found for the track. This approach works well for muons but not for electrons, which have a short track due to electromagnetic showering. The second point is found by projecting all PMT charges on the track at the same Cherenkov angle. The fraction of the charge between the beginning of the track and the PMT determines the fraction of the total track length for the PMT. The total track length is assumed to correspond to an energy loss of 3 MeV/cm. The shortest of the two distances is chosen for each PMT.

The final step is the adjusting of the vertex position. This is done by minimizing the width of the residual time. For PMTs outside of the Cherenkov cone, light is assumed to come from the vertex, for PMTs inside of the cone, light is assumed to come from the calculated point along the track. For PMTs outside of the cone a provision is made to allow for the long timing tail due to scattered light.

5.2 Ring Finding

The ring finding algorithm consists of two steps repeated iteratively. The first step finds the possible ring candidates (directions). The second step tests the candidates to determine whether they are real rings or not. This process is repeated until no more candidates are found, but no more than four times.

5.2.1 Finding ring candidates

The finding of the candidates is based on the Hough transform [48]. This transformation is especially well suited for finding narrow, sharp rings of a fixed radius. For every point on an observed circular pattern of a given radius a circle is drawn with the same radius. All the drawn circles are added together. They all add up (intersect) at the center. Ring candidates are found by histograming and adding the contributions from all the drawn rings and looking for peaks in the histogram distribution. The Hough space histogram is binned in spherical coordinates θ (36 bins) and ϕ (72 bins), the PMTs are viewed from the reconstructed vertex. Before histograming, charge from already reconstructed rings is subtracted in order to prevent strong (or close) rings from overwhelming weaker rings. Charge detected by every PMT is corrected for attenuation and acceptance. It is mapped to Hough bins which are at the Cherenkov angle from the PMT. To account for fuzzy rings, the PMT charge is spread over a wider range of angles, corresponding to the charge distribution of a 500 MeV electron. Peaks in the Hough histogram are identified and those which are closer than 15 degrees from already reconstructed ring are rejected. To account for varying ring opening angles due to collapsing Cherenkov rings of low momentum particles or due to error in vertex reconstruction, the Hough histogram is made with several opening angles, and the one with the sharpest peak is selected.

5.2.2 Testing Ring Candidates

For this test, the likelihood that an observed light distribution matches the hypothesis of a given number of rings is calculated. Ring charge separation is used in the process.

First, a likelihood with the new ring candidate is calculated. Only PMTs which are within the angle of the Cherenkov cone multiplied by 1.2 to any of the rings are used. A 500 MeV electron expected charge distribution is used for the new ring.

For the other rings, the expected charge contributions are iteratively optimized to maximize the overall likelihood. The new ring momentum is constrained to prevent it from getting too low.

Next, a likelihood with only already reconstructed rings is calculated. Exactly the same PMTs are used as in the previous likelihood. The expected charge distributions are fixed, but the coefficients of how much each of them contributes are optimized.

The two likelihoods are compared. If the likelihood with the new ring is lower than the likelihood without it, the ring is rejected.

Several functions are then calculated. The larger they are the more probable the ring candidate is. They are multiplied (with some empirical weights) to get a final function which is then evaluated to make a final decision whether the new ring is accepted or rejected.

The first function is the difference of the new likelihood with the new ring minus the likelihood without it with a total charge correction applied.

The second function is a sum of charge near the edge of the new ring candidate which is assigned to this ring in regions which do not overlap with already reconstructed rings.

The third function adds all the charge assigned to the new ring which is outside of the Cherenkov cone in regions not covered by other Cherenkov cones. The final function is the difference of the second function and the charge sum, which is corrected for total charge and for opening angles.

The fourth function is a total amount of residual charge for already reconstructed rings (observed charge minus expected charge), corrected for the total amount of charge.

5.2.3 Ring Separation

The ring separation (charge separation) routines separate observed charge in the event, assigning portions of it to each of the reconstructed rings. Each PMT hit is assigned how much of light was due to each ring. This is done by assigning to each ring an azimuthally (ϕ) symmetric light distribution as a function of polar angle θ. Using this, one can calculate how much light was emitted due to any PMT from this ring. Having this distribution, the fraction of charge at each PMT due to a given ring is calculated by dividing expected light due to this ring by total expected light due to all rings.

The heart of the charge separation routines is the calculation of the light distribution as a function of angle θ for each ring, including normalization of each ring. The process consists of iterations of generating light distribution and recalculating ring weights. Four iterations are performed.

At first, each ring is assumed to generate an electron-like light distribution. Then, weights are optimized by minimizing χ^2 using only PMT hits within a 70 degree cone of a ring. This completes the first iteration.

At this step, an improved light distribution is calculated. This is done by adding the separated charge of PMT hits, which is attributed to each of the rings. The charge is binned in θ bins. The adding is modified in such way, as to ensure that most of the contribution comes from non-overlapping regions of the rings. Charge is corrected for attenuation, distance and PMT acceptance. It is also smoothed.

Then, ring weights are calculated a second time. This time, a likelihood is maximized. In the likelihood contributions from the ring edge and from within the Cherenkov cone are enhanced. This completes the second iteration.

The third iteration repeats steps of second iteration, but with diffusion of Cherenkov light taken into account. The fourth iteration repeats the same procedure, but with the finite size of the PMTs taken into account.

5.3 Particle Identification

This part of the reconstruction determines particle type (often called particle ID or PID): showering (e-like, mostly electrons and gammas) or non-showering (μ-like, mostly muons and charged pions).

First, the ring separation is performed in order to get the separate observed charge distribution for each ring. Then the observed charge distribution for each ring is compared to the expected charge distribution for each particle type.

The showering particle expected light distribution is tabulated as a function of opening angle θ. The table was built by Monte Carlo simulation of electrons at three momenta: 100 MeV, 300 MeV and 1000 MeV. The light distribution at 16.9 m from the vertex was used. The light distribution is interpolated (and extrapolated) to obtain distributions for a wide momentum range. The expected charge for each PMT is corrected for the opening angle of the PMT as seen from the vertex, for attenuation length, for PMT angular acceptance, and for diffusion of the Cherenkov light.

The expected light for non-showering particles is calculated analytically. The direction in which Cherenkov photons are emitted as the muon loses energy is used, and the intensity of Cherenkov light as a function of Cherenkov angle is used. A term based on Monte Carlo is added to account for knock-on electrons. The expected light is corrected for attenuation and PMT acceptance.

Scattered light is added to both the showering and non-showering expected light distribution, Observed PMT hits are classified as direct light hits or scattered light hits based on their residual timing. Late hits are assumed to be due to scattered light. Based on this, a scattered light distribution is estimated.

Using the expected light distributions, the particle type of each ring is determined in sequence. While determining a particle type for a given ring, the expected charge for previous rings is calculated using the charge separation of the observed charge. The expected charge for the ring considered is the (non-)showering expected light distribution described above.

A total likelihood is calculated by multiplying likelihoods of individual PMTs. Only PMTs which are within the opening angle of the current ring multiplied by 1.5 are considered. The likelihood is then maximized by changing the direction of the current ring and its opening angle.

Using these two likelihoods, probabilities that the ring is due to an electron or due to a muon are calculated.

A second set of probabilities is calculated based only on the opening angle information. A combined probability is then calculated by multiplying the opening angle probability by the pattern probability.

For single ring events, if the probability of electron is greater than the probability of muon, then the event is labeled electron-like. Otherwise, it is muon-like.

5.4 MS Vertex Fit

An additional adjustment of the vertex is performed for single ring events. This fitter uses PMT timing, expected charge distribution using the reconstructed particle type, and estimated momentum. Vertex position and ring direction are varied. The Cherenkov angle is kept fixed—this gives a strong constraint in the vertex position along the track direction. PMT timing alone gives only relatively weak constraint in this direction for many event geometries.

5.5 Momentum Determination

For multiple ring events charge separation is performed. The total corrected charge for each ring is calculated. Only PMTs within a 70 degree opening angle are used for each ring. Also, a time residual cut is applied from -50 to 250 ns, measured from the peak of the residual time distribution. Each PMT hit is corrected for attenuation, angular acceptance and PMT coverage. The total charge is corrected for PMT gain variation with time (calculated using cosmic ray through going muons). The corrected charge then is converted to momentum assuming that the particle is either electron or muon, as dictated by the reconstructed particle type.

5.5.1 Ring Separation

Charge is separated using determined particle type and an estimated momentum. The expected light distribution used here is tabulated from Monte Carlo simulation. Scattered light and reflected light are taken into account. Ring weights are optimized by maximizing a likelihood.

5.6 Decay Electrons

There are three kinds of muon decays observed in Super-Kamiokande. PMT hits from decays within 0.9 μs from main event are contained in the main event PMT hit list. They are called “in-gate” decays. Decays later than 0.9 μs generate a separate trigger and are contained in an event of their own. Those events are attached to main events during reduction. Such decays are called “after-event” or “sub-event” decays. The final case is when muon decays around 0.9 μs after the main event. Some of the PMT hits are contained in the main event, and some are contained in a separate after-event. Those events are often called “bye-bye” decays.

5.7 Ring Number Correction

For events with 3 or more rings, each ring is tested to see if it is a real ring. Any ring is rejected if one of two conditions is fulfilled. The first condition tests the given ring with each of the remaining rings. The ring is rejected if it’s visible energy is smaller than the other ring, if the opening angle between the ring is smaller than 30 degrees, and if the visible energy of the first ring multiplied by the cosine of the opening angle between the rings is smaller than 60 MeV. The second condition rejects the ring if its visible energy fraction compared to the total visible energy is smaller than 0.05 and if its visible energy is smaller than 40 MeV.

Chapter 6

Monte Carlo Simulation

Three major steps are involved in the Monte Carlo simulation [49, 50]: calculation of the neutrino flux, neutrino interactions in water, and detector simulation.

6.1 Neutrino Flux

Neutrino flux is calculated by following the development of cosmic ray showers in the atmosphere, starting with the primary cosmic ray flux. It is calculated specifically for the Super-Kamiokande site.

6.1.1 Primary Cosmic Rays

The primary cosmic ray flux is calculated separately for low energies and for high energies. The boundary is at about 100 GeV (neutrino energies of about 3–10 GeV). The composition of the cosmic rays is measured to be 95 percent H, 5 percent He, and less than 1 percent CNO, which corresponds to 80, 15, and 5 percent of neutrino flux, respectively.

Lower energy cosmic rays, below about 10 GeV, are affected by solar activity. The cosmic ray flux decreases when solar activity increases. Solar activity is measured by sunspot number and by measuring cosmic ray proton, helium and neutron rates [51, 52]. Cosmic ray flux was near that of solar minimum since 1996 (when data taking started) until summer 1999, decreasing during the next year, and was near that of solar maximum from summer 2000, until end of data taking in July 2001. For this reason, Monte Carlo is calculated for 3 years of solar minimum, 1 year of changing activity, and 1 year of solar maximum. The rate of sub-GeV events at solar minimum is 2 percent higher than average.

The lowest momentum cosmic rays, below about 10 GeV, are also affected by the Earth's magnetic field. Some of these primary cosmic rays are bent by the field and never reach the atmosphere. On the Earth's surface, for any direction, there is a minimum rigidity (momentum per unit charge) below which no primary cosmic rays arrive. This cutoff rigidity is calculated by tracing antiparticles backwards though the magnetic field and rejecting those whose orbit intersects Earth, as shown on Figure 6.1. Particles which reach 10 Earth radii where the magnetic field of the Earth has the same strength as the ambient field in the Solar System are accepted. The cutoff rigidity is higher for the eastern direction than for the western direction. This east-west effect in the neutrino flux was observed for the first time by Super-

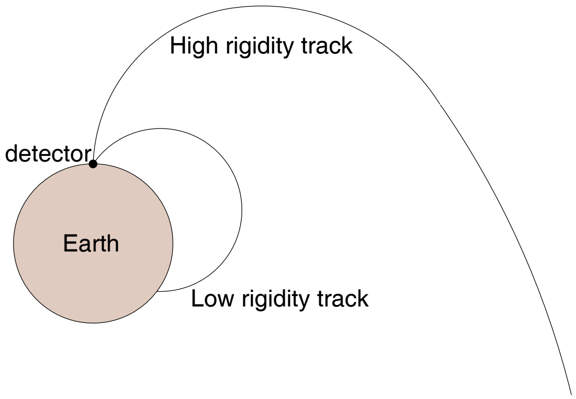


Figure 6.1: Schematic illustration of rigidity cutoff.

Kamiokande [53]. Its observation confirms the validity of the geomagnetic effects used in the neutrino flux calculation.

The high energy primary cosmic ray flux is expressed as a power law.

6.1.2 Atmospheric Showers

A primary cosmic ray collides with a nucleus of atmospheric gas and produces a cascade of secondary particles. The density and the composition of the atmosphere used for the calculation is taken from standard tables. The most frequent interaction of the primary cosmic ray is π^+ production. This introduces a neutrino-antineutrino asymmetry, which can be traced back to the matter-antimatter asymmetry of the primary cosmic rays.

The most detailed atmospheric neutrino flux calculations [54, 55, 56, 57, 58] are fully three dimensional calculations, unlike the older one dimensional calculations [59, 60] that assumed that secondary particles travel in the direction of the primary

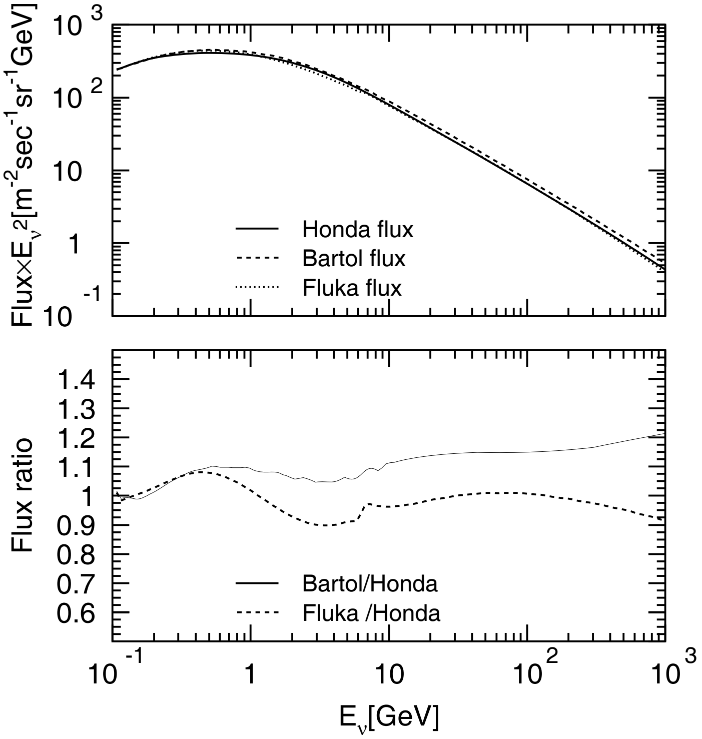


Figure 6.2: Top: spectrum of $\nu_\mu + \overline{\nu}_\mu$, averaged over all directions, Calculations reported in [55] (solid line), [56] (dashed line), and [58] (dotted line). Bottom: Ratios of the above neutrino fluxes.

cosmic ray. The calculated flux is shown on Fig. 6.2.

6.2 Neutrino Interactions

Neutrino interactions in water are modeled by specialized programs developed at Super-Kamiokande. Neutrinos scatter on protons, oxygen nuclei and electrons. The interactions on electrons at typical atmospheric neutrino energies can be neglected. Nuclear effects such as Fermi momentum, Pauli blocking, deexcitation, and propagation in the nucleus are taken into account.

6.2.1 Quasielastic and Elastic

Quasielastic scattering is a charged current reaction in which the incoming neutrino produces a corresponding lepton and the nucleus transforms according to charge conservation, for example:

$$\nu_\mu + n \rightarrow \mu^- + p \tag{6.1}$$

$$\overline{\nu}_\mu + p \rightarrow \mu^+ + n \ . \tag{6.2}$$

Elastic scattering on a nucleus is a neutral current reaction in which particles do not change their identities.

Quasielastic scattering off free proton is described by Llewellyn-Smith [61]. For scattering off nucleons in ^{16}O, the Fermi motion of the nucleons and Pauli exclusion principle are taken into account. The nucleons are treated as quasi-free particles using the relativistic Fermi gas model of Smith and Monitz [62]. The momentum distribution of the nucleons is assumed to be flat up to the fixed Fermi surface momentum of 225 MeV. This Fermi momentum distribution is also used for other nuclear interactions. The nuclear potential is set to 27 MeV. Pauli blocking is implemented by accepting only interactions in which the final state nucleon has a momentum above the Fermi level. The neutral current cross-section is derived from the charged current cross-section.

Elastic neutral current interactions are rarely seen in practice in Super-Kamiokande. In order to be visible, the recoil proton has to be above Cherenkov threshold, or the recoil nucleon has to produce visible pions.

6.2.2 Resonant Meson Production

At higher energies, resonances become significant. The excited nucleon then decays into a nucleon and a meson. For example:

$$\nu_\mu + p \to \mu^- + \Delta^{++} \tag{6.3}$$

$$\Delta^{++} \to p + \pi^+ \ . \tag{6.4}$$

The Rein-Seghal model is used to simulate resonant production of single π, K, and η [63, 64, 65]. Mesons K, and η are important in order to accurately predict proton decay backgrounds. We calculate the amplitude of producing each resonance. We multiply each amplitude by the probability the corresponding resonance decays into one meson. The hadronic invariant mass W (here, the mass of the intermediate resonance) is restricted below 2 GeV for consistency with multi-meson production (deep inelastic scattering is used above this value). Interference among the constituents is taken into account.

The strongest resonance is $\Delta(1232)$ P_{33}. The pion angular distribution for the decay of this state is calculated using Rein-Seghal's method. For the remaining resonances, the directional distribution of the generated pions is set to be isotropic in the resonance rest frame. The angular distribution of π^+ has been measured for

$$\nu_\mu p \to \mu^- p \pi^+ \tag{6.5}$$

mode [66] and the results agree well with the Monte Carlo prediction.

We also consider the Pauli blocking effect in the decay of the baryon resonance by requiring that the momentum of the nucleon should be larger than the Fermi surface momentum. Pion-less delta decay is also considered, where 20% of the events do not have the pion and only lepton and nucleon are generated [67].

The quasielastic and single meson production models have a parameter, axial vector mass M_A that must be determined by experiments. For larger M_A values, interactions with higher Q^2 values (and, therefore, larger scattering angles) are enhanced for these channels. The M_A value was tuned using the K2K [30] near detector data. In the atmospheric neutrino Monte Carlo simulation, M_A is set to 1.1 GeV for both the quasielastic and single meson production channels.

6.2.3 Coherent Pion Production

Single pions are also produced in coherent pion production on the oxygen nucleus, and in diffractive pion production on protons, for example:

$$\nu_\mu + {}^{16}\mathrm{O} \to \nu_\mu + \pi^0 + {}^{16}\mathrm{O} \ . \tag{6.6}$$

In such processes, the nucleus recoils as a whole. Because the momentum transfer in such interaction is small, their angular distribution is more forward-peaked than that of pions from resonances, i.e., their direction is more correlated with the direction of the incoming neutrino.

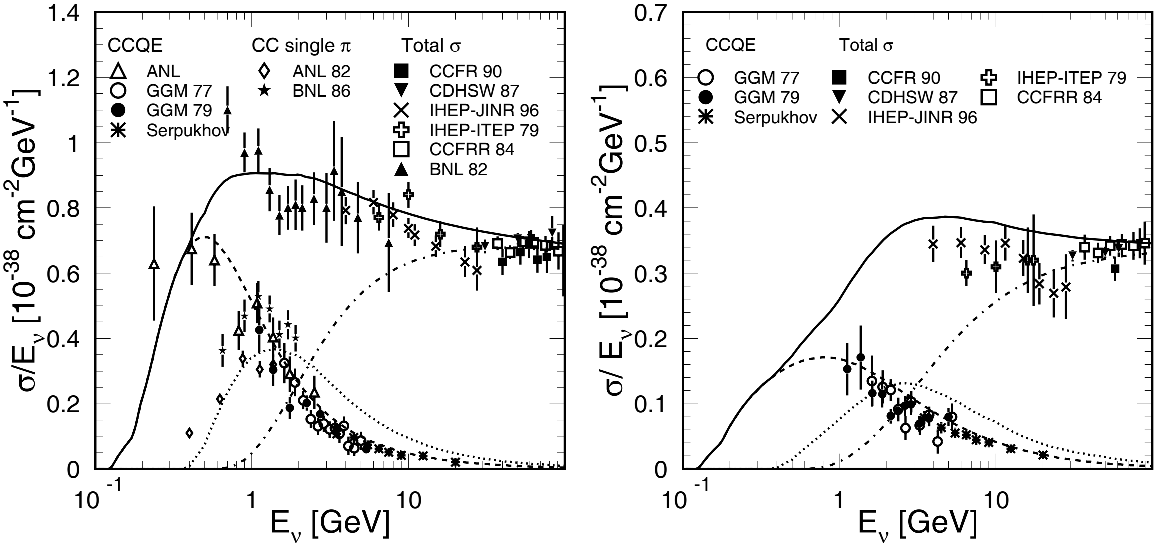


Figure 6.3: Charged current total cross-section divided by E_ν for neutrino (left) and anti-neutrino (right) nucleon charged current interactions. Solid line shows the calculated total cross-section. The dashed, dotted, and dash-dotted lines show the calculated quasi-elastic, single meson, and deep inelastic scattering, respectively. Data points are from ANL [72], GGM 77 [73], GGM 79 [74, 75], Serpukhov [76], ANL 82 [77], BNL 86 [66], CCFR 90 [78], CDHSW 87 [79], IHEP-JINR [80], IHEP-ITEP 79 [81], CCFRR 84 [82], and BNL 82 [83].

6.2.4 Deep Inelastic Scattering

In order to calculate the cross-sections of the deep inelastic scattering, the GRV94 [68] parton distribution function is used. In the calculation, the hadronic invariant mass, W, is required to be greater than 1.3 GeV. However, the multiplicity of pions is restricted to be 2 or greater for $1.3 < W < 2.0$ GeV, because single pion production is separately simulated, as described before. In order to generate the events with multi-hadron final states, two models are used. For W between 1.3 and 2.0 GeV, a custom-made program [69] is used to generate the final state hadrons. For W larger than 2 GeV PYTHIA/JETSET [70, 71] is used.

The total charged current cross-sections, including quasi-elastic scattering, single meson production, and deep inelastic scattering are shown in Fig. 6.3.

6.2.5 Nuclear Effects

Produced mesons propagate through the ^{16}O nucleus. Their interactions are treated using the cascade model. The interactions of pions are very important, because the cross section for pion production is quite large for neutrino energies above 1 GeV and the interaction cross-sections for pions in nuclear matter is also large.

In the simulation program, the following pion interactions in ^{16}O are considered: inelastic scattering, charge exchange, and absorption. The actual procedure to simulate these interactions is as follows. The generated position of the pion in the nucleus is set according to Woods-Saxon nucleon density distribution [84]. The interaction mode is determined by using the calculated mean free path of each interaction. To calculate the mean free path, model described by Salcedo *et al.* [85] is used. The calculated mean free paths depend not only on the momentum of the pion but also on the position of the pion in the nucleus. If inelastic scattering or charge exchange occurs, the direction and momentum of the pion are determined by using the results of phase shift analysis obtained from $\pi - N$ scattering experiments [86]. When calculating the pion scattering amplitude, the Pauli blocking effect is also taken into account by requiring the nucleon momentum after interaction to be larger than the Fermi surface momentum at the interaction point. The pion interaction simulation was checked using data for π ^{12}C scattering, π ^{16}O scattering, and pion photo-production [87]

$$\gamma + {}^{12}\mathrm{C} \rightarrow \pi^{-} + X \ . \tag{6.7}$$

6.3 Detector Simulation

Information produced in the neutrino interaction (particle types and their momenta) is passed to the detector simulation. It simulates the propagation of particles in water, their interaction and the production of secondary particles, the production of Cherenkov light, light propagation in water (including scattering, absorption and reflection), the PMTs, and the electronics response.

6.3.1 Particle Propagation

Particles are tracked using a Monte Carlo program based on the GEANT package [88] with custom software modules. Tracking includes interactions, the production of secondary particles, and the decays. Hadronic interactions in water above 500 MeV are handled in the simulation by CALOR package [89]. This package is known to reproduce the pion interactions well including low energy region ($\sim$ 1 GeV). For still lower energy region ($p_\pi \leq 500$ MeV), a custom program based on experimental data from $\pi - {}^{16}$O scattering [90] and $\pi - p$ scattering [91] is used in the simulation code.

6.3.2 Cherenkov Light

Cherenkov photons are generated for every 1 cm step of travel of charged particles, distributed uniformly along the track. Light is propagated with speed dependent on wavelength.

Light scattering and absorption coefficients used in the simulation are tuned to reproduce laser measurements in the detector (Section 3.4.1). For short wavelengths, Reighleigh scattering dominates. This is scattering on small particles (r much smaller than λ) and on water molecules. For long wavelengths, absorption dominates. Forward peaked Mie scattering on large particles (r much greater than λ) is also simulated.

Light reflection and absorption on the detector material such as the black sheets and Tyvek in the detector were measured and are included in the simulation. Measured PMT reflectivity is also simulated. These distributions depend on photon incident angle.

6.3.3 Electronics

In simulating the detection of light by a PMT, a measured quantum efficiency is used. Charge which is too high is truncated to simulate saturation. Time and charge are smeared to simulate measurement resolution. Dark noise is added with the rate measured in Super-Kamiokande. In later analysis, bad PMTs are simulated by applying a mask.

Chapter 7

Tscan

Tscan ("**T**rivial **Scan**ner") is an event display, traditionally called a scanner, which I developed. It is a program that shows events graphically on the computer screen.

It was designed to be simple ("trivial") internally, and to have a simple user interface. A lot of importance was given to giving the user a large choice of options to display events in many different ways.

Tscan proved to be a very useful tool for the development of fitters. A particularly useful feature is the ability to show custom data for every PMT. Instead of the usual time and charge, it can show expected charge, scattered light, likelihood, χ^2 difference, patches (see Figure 8.3), and any other data that can be prepared in a text format.

7.1 Features

A detailed description of features, command line options, and keystrokes is contained in the manual page and on the World Wide Web [92]. Example images can be found there too. Some of the features are highlighted below.

7.1.1 Image Features

The flexibility of many ways to display an image was a major design goal.

The three-dimensional detector is shown in several ways projected onto a two-dimensional display. One way, is the standard unrolled cylinder image with side wall shown as a rectangle, and top and bottom walls shown as circles above and below the rectangle. There are several ways of showing a spherical image, as seen by an observer, onto a plane. Finally, there is a realistic perspective view and a stereo view. The view can be switched to the outer detector.

PMTs can be displayed in several ways: as a points, as a circles of fixed size, as circles with size adjusted depending on distance from observer, and, most realistically, as flat squares in three-dimensional space. They can be drawn as an outline, or as a filled shape.

Usually PMTs with small amount of charge are drawn small, and “hot” PMTs are drawn larger, but size can depend on other quantities as well. The function of translation of charge to linear size can be linear, square root (linear area), or logarithmic (for large dynamic range).

To enhance the perception of location within the detector and the sense of detector shape, all non-hit PMTs can be shown in gray, or detector shape can be outlined with a help of grid pattern (with adjustable grid density). Additionally two kinds of grid can be drawn centered on the observer point, which helps one to understand the projection of a spherical view onto the plane (Fig. 7.1 and Fig. 7.2).

PMT colors can represent charge, time or two kinds of time of flight subtracted (residual) time. One kind of residual time assumes that all light arrives from the vertex (point residuals) the other kind assumes that light arrives at the Cherenkov angle from a linear track (long residuals). A histogram of the quantity used for color scale is displayed, with hits weighted by amount of charge (except in charge histogram). To improve print output in some cases, the image can be drawn in reverse video, i.e., with light background and dark lines. In addition, color scale can be replaced with gray scale to improve black and white printouts. The image can be saved to a file for printing.

Tracks contained in the data file (for example, reconstructed tracks, or Monte Carlo tracks) are shown primarily as continuous lines drawn where Cherenkov cone intersects cylindrical walls. Additionally, the projected entry and exit points are marked (extending the track forward and back) and perpendicular projections of the vertex to the nearest top, bottom and side walls are shown.

With single keystrokes the user can move within the detector, rotate the view, and also shift, stretch, and compress the color scale.

7.1.2 Other Features

The data can be read in three formats: old offsite zform format, current ZBS format, and in text format based on superform format. In the first two formats, the user can move in both directions, forward and back.

Details of the event information, including tracks and Monte Carlo information, can be displayed.

The exact state of the scanner can be saved to a file, so that later user can resume event viewing in exactly same state.

7.2 Under the Hood

Tscan is written in the C language. Graphical features are implemented using raw `Xlib` (a very basic, low level library for X-Windows) without any toolkit on top of it. This allows flexibility and avoids toolkit limitations.

The main loop consists of waiting for X-Windows events (key presses, window resizing, etc.) and reacting to them. The central routine is `draw_detector()`, which draws all the elements of the image (except the color scale and histogram), using current projection and other settings.

In order to be independent of the input data format, all PMT hit information is stored in a custom structure. It is easy to add more data formats by adding custom read routines in `read_format.c` files. Internally, the detector geometry is stored in the old offsite `skgeom` structure, even when it is read from current format.

Most program settings are stored in user options structure `uopt`. These settings are written out to a file when saving program status. The file is just a shell script containing the `tscan` command with many options, one option for each setting.

A special data type `enum` was implemented to store a setting which can take several string values. Functions are provided for setting, reading, testing, incrementing and decrementing (modulo number of possible settings) of the `enum` type.

Amount of debug text output is adjustable with `tdebug` variable.

By default (except in perspective view), the initial view position is selected at the first vertex. In such a case, the initial view direction is based on the charge distribution around the vertex. The charge inertia tensor, where charge is projected onto a sphere and is treated like mass, is diagonalized (see Section 8.3.1). Its eigenvectors are used as principal axes of the view in such way that the viewer is facing the maximum charge direction, maximum charge spread is in the horizontal direction, and the distribution of charge in the vertical direction is smallest.

In the perspective view, the initial position is selected in a corner of the detector farthest from the majority of the charge, and view direction is toward the center, at angle of 45 degrees with respect to horizontal plane.

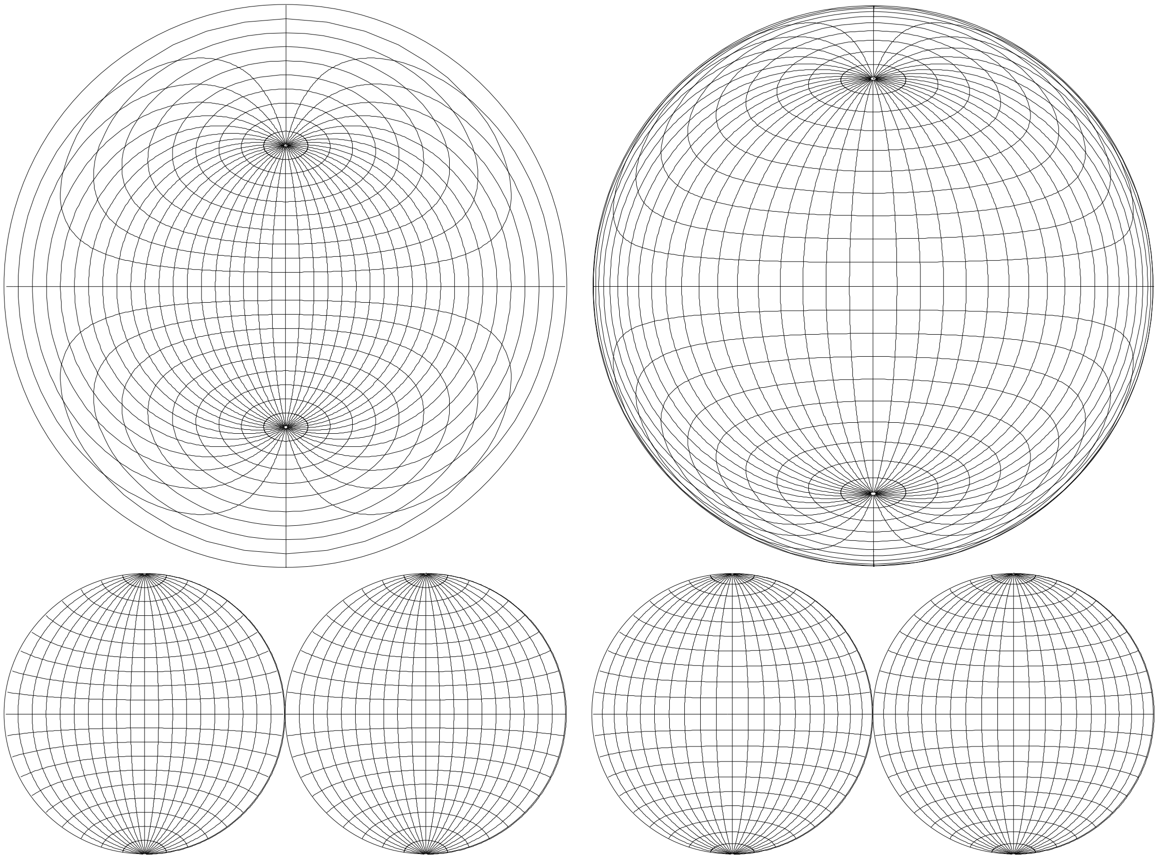


Figure 7.1: Projections of tscan. Top row shows spherical projections. Bottom row shows hemispherical projections. Left side shows non equal area projections. Right side shows equal area projections. Lines of constant longitude and latitude are drawn every 10 degrees.

7.2.1 Projections

When writing tscan, a major objective was to support many different projections and to make it easy to add new projections as needed. A projection is a way of showing points in three dimensional space on a two dimensional screen. Drawing of any three dimensional object can be easily implemented, for example, if needed, drawing of the linac pipe, calibration sources, or the water pipes could be added. Examples of objects already implemented are Cherenkov cones (or, more precisely,

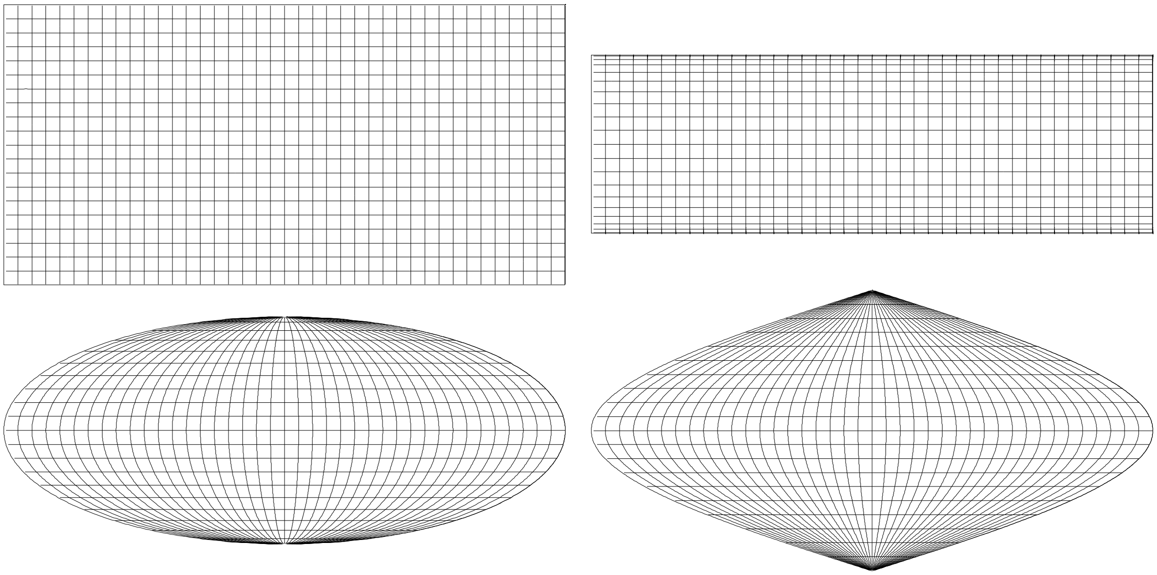


Figure 7.2: Projections of tscan. Top-left: non equal area cylindrical projection. Top-right: equal area cylindrical projection. Bottom-left: Mollweide globe projection. Bottom-right: sinusoidal globe projection. Lines of constant longitude and latitude are drawn every 10 degrees.

their intersection with detector walls) and lines outlining shape of the detector when drawing detector grid. All those lines are curved lines (series of straight line segments) in three dimensional space.

7.2.1.1 Steps

A projection is implemented as a two-step process. First, the most important step, is the translation of the three dimensional coordinates into a convenient set of two dimensional coordinates, which is independent of window size. The names of functions which perform this transformation start with `r2f` (real-to-flat). The second step is the transformation of those convenient two dimensional coordinates into window coordinates. The names of functions which perform this transformation start with `f2w` (flat-to-window). In most cases the two steps are combined by "wrapper" functions which perform the full transformation. Names of these functions start with `r2w` (real-to-window).

Each set of real-to-flat functions resides in its own `r2f` file. All data and functions in the file are private (static in C language terminology) except a structure which provides pointers to three interface functions. Adding a new projection involves, essentially, not much more than adding a new `r2f` file with the three functions. One of the functions is `init_real_to_flat()`, which informs the external code about the scale of the flat coordinates used, and initializes some internal data. Another is `draw_outlines()` which calls functions which draw objects in flat coordinates (circles, rectangles, curves, etc.). This draws an outline of the projection. No three dimensional point ever gets projected outside of the outline in the flat space. The final, most important, function is `real_to_flat()`, which performs the actual projection transformation.

Before performing the projection, the detector is rotated, to account for the orientation of the observer. In an unrolled view a simplified rotation is performed, limited to rotation around the z axis only.

Some of the projections are equal area projections. This means that an area of a sphere of a given size is always shown as an area on the projected image of the some fixed size, regardless of the direction where the area is located.

7.2.1.2 Unrolled Projection

In the unrolled projection, the side wall appears as a rectangle, which comes from cutting and unrolling the detector cylindrical side wall, and top and bottom walls appear as circles at the top and bottom of the rectangle. This projection can be seen on Figure 8.3. This view differs from the other views in that it does not depend on the location of the observer. The view is always relative to the detector center.

First, a decision is made as to whether the point belongs to top, side, or bottom wall, based on the angle of the point location measured from detector center with respect to the x axis. For increased speed, this information can be passed from the calling routine, in which case this step is skipped.

A position on the side wall is determined by its ϕ and z coordinates. A position on the top or the bottom walls is determined by its x and y coordinates,

The unrolled view is useful for seeing how and where the hit pattern is distributed within the detector.

7.2.1.3 Cylindrical Projection

In this projection (Fig. 7.2 top-left, and Fig. 7.3), the viewer is assumed to be at the origin of a spherical coordinate system, facing direction $\phi = 0$ with the equator being horizontal. The coordinate ϕ determines horizontal position, while coordinate θ determines vertical position on the displayed image. The viewing area is a rectangle with coordinates in flat system ranging from $-\pi$ to π horizontally and from $-\pi/2$ to

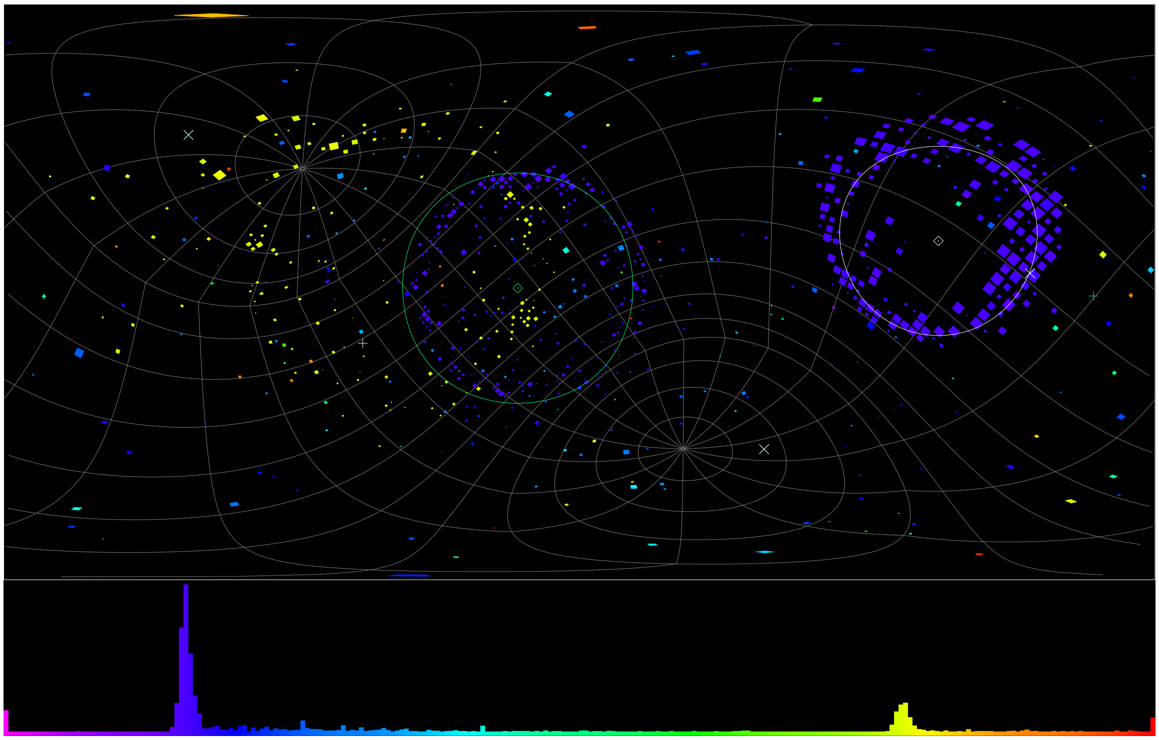


Figure 7.3: A two ring event with muon decay electron in cylindrical view. The viewer is auto-positioned in the reconstructed vertex. Grid shows outline of the detector, with the top of the detector in upper left, and bottom of the detector in center bottom. PMT size shows charge, color shows time of flight (from the reconstructed vertex) subtracted (residual) time. A time of flight cut (600 ns wide, the histogram range) was performed to remove PMTs that were hit too early or too late. The histogram at the bottom (above a base line shown for color scale) shows charge as a function of residual time (600 ns span, extreme bins are overflow bins). Two peaks are seen in the histogram: one for the main two rings (rings visible in blue, in center (91 MeV) and right (72 MeV), shown with projected reconstructed Cherenkov cone outlines, on detector side wall), and another for electron from muon decay, 370 ns later (fuzzy ring visible in yellow, in center left, overlapping ∼ 50% of the center ring, partially on detector top wall and mostly on detector side wall, without Cherenkov cone outline shown). X marks show reconstructed vertex projections on walls. Diamonds mark projected reconstructed track exit points, and plus signs mark back projected track entry points. This event was recorded on 1996/05/29 21:23:05 (run 1734 subrun 6 event 38449).

$\pi/2$ vertically. Distortion is small near the equator (middle horizontal region of the viewing area), and large near the poles (top and bottom of the viewing area).

In a second version of this projection (Fig. 7.2 top-right), vertical position is determined by $\sin\theta$ instead of θ. Vertical viewing area is smaller, it ranges from -1 to 1. This is an equal area projection. However, distortion of shapes near the poles is greater than in the first projection.

The cylindrical projection is good for viewing planar events, for example two ring events or three ring events with small total momentum, as would be expected in many nucleon decay events.

7.2.1.4 Spherical Projection

Here, the viewer is assumed to be in a spherical coordinate system, facing the direction of the pole $\theta = 0$ (non-standard θ definition, ranging from 0 to π instead of the usual $-\pi/2$ to $\pi/2$). Spherical coordinates are translated into polar coordinates with spherical θ being polar r and spherical ϕ being polar ϕ. The viewing area is a circle with radius of π in the flat coordinate system. The least distorted area is at the center of the image (front pole), and largest near the edge (back pole). This projection is shown on Fig. 7.1, top-left. Note, that the figure shows grid with poles on top and bottom, not in front and behind the viewer.

In equal area variant of this projection (Fig. 7.1 top-right) polar r is sine integral of θ. The viewing area circle radius is sine integral of π, approximately 1.85194.

7.2.1.5 Hemispherical Projection

This projection is similar to the spherical projection. The difference is that the image is split into two hemispheres. In the forward hemisphere (shown as circle of radius $\pi/2$) θ between 0 and $\pi/2$ is shown. A second circle of same radius is drawn on the left side, showing θ between π and $\pi/2$ with π (backward pole) being shown at the center of the circle and $\pi/2$ (equator) being shown the the edge of the circle. This projection is shown on Fig. 7.1, bottom-left. Note, that the figure shows grid with poles on top and bottom, not in front and behind the viewer, just like in case of spherical projection.

This projection avoids large distortion near the back pole, but the price paid is having an image in two discontinuous pieces. The largest distortion is at the edges of the viewing areas, while the centers of the viewing areas are the least distorted. It is good for viewing back-to-back events as would be expected for many nucleon decay events. This projection is also available in equal area version (Fig. 7.1 bottom-right).

7.2.1.6 Globe Projections

This projection is used frequently to map the surface of the Earth, and for sky maps. Like in the cylindrical projection, the viewer is placed in a spherical coordinate system with poles above and below him. There are two variants of the projection, both of which are equal area projections.

Sinusoidal projection (Fig. 7.2 bottom-right) is the simpler of the two. The vertical position of a point is simply the θ coordinate, and ranges from $-\pi/2$ to $\pi/2$. Angle ϕ, which ranges from $-\pi$ to π, determines horizontal position. In order to achieve equal area projection, before plotting, the horizontal coordinate is multiplied by $\cos\theta$. The width of the viewing area changes with angle θ. This gives the viewing area a characteristic rounded shape, elongated horizontally, with pointed top and bottom, where the poles reside.

The Mollweide projection [93] (Fig. 7.2 bottom-left) is more complex. The horizontal coordinate is

$$x = \phi \cos\theta_{\mathrm{M}} \ , \tag{7.1}$$

the vertical coordinate is

$$y = M \sin\theta_{\mathrm{M}} \ , \tag{7.2}$$

where θ_{M} is given by

$$2\theta_{\mathrm{M}} + \sin\theta_{\mathrm{M}} = \pi \sin\theta \ , \tag{7.3}$$

and M is a constant. θ_{M} is calculated iteratively using Newton's method. In this projection, the viewing area has an oval shape without sharp corners.

In the above two projections distortions are small at the center and close to the equator and are largest at the edges, particularly near the poles.

7.2.1.7 Perspective Projection

Technically, this projection is somewhat similar to the (hemi)spherical projection. The observer is assumed to be located in a spherical coordinate system, facing the pole

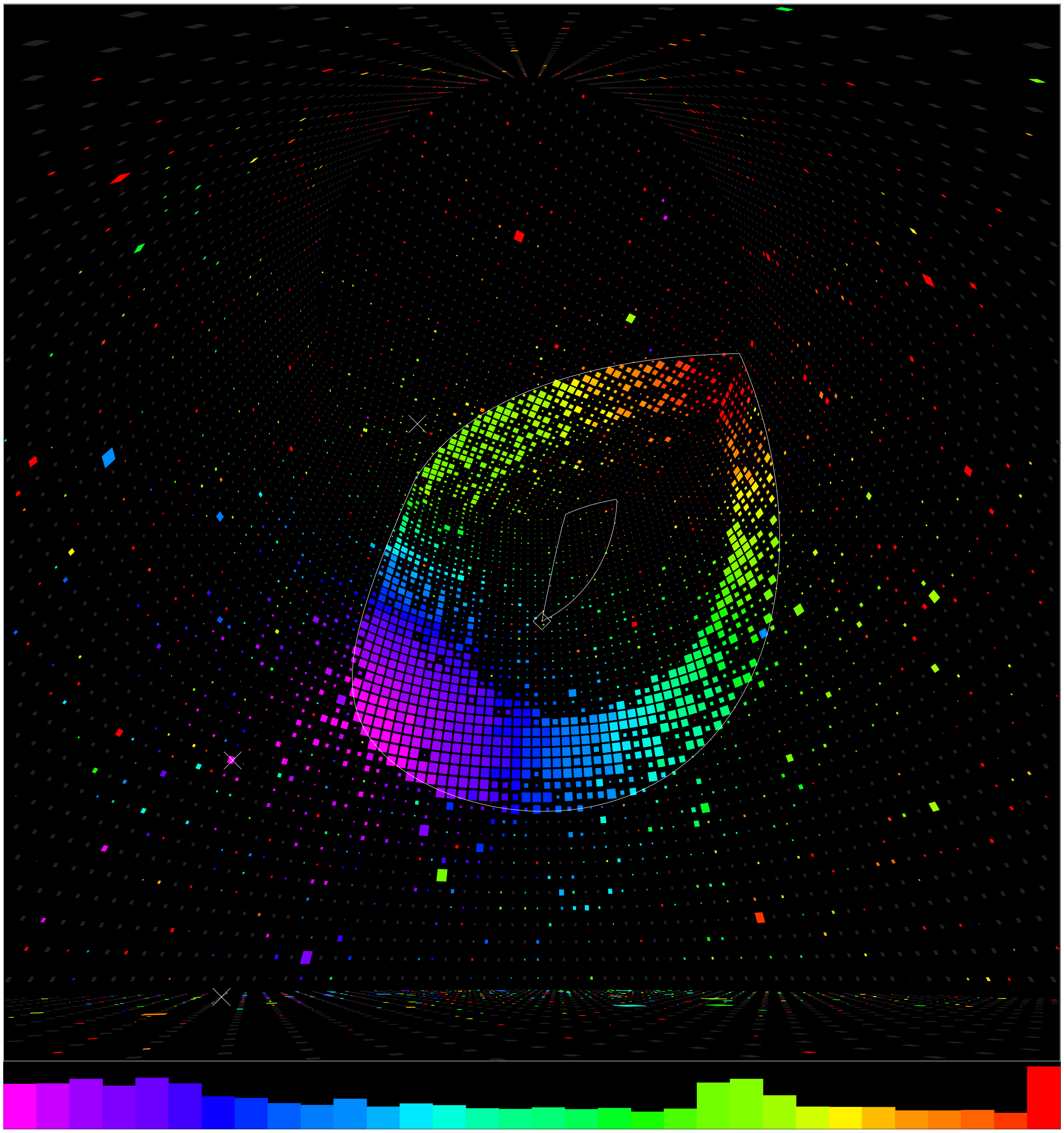


Figure 7.4: A single ring event (muon-like) in perspective view. Viewer is auto-positioned in far corner. PMT size shows charge, color shows raw time (i.e., no time of flight correction was applied). The histogram at the bottom (above a base line for color scale) shows charge as a function of raw time (120 ns span). X marks show reconstructed vertex projections on walls. Diamond shows projected reconstructed exit. Outline shows projected reconstructed Cherenkov cone. Inner outline shows where the cone would end if the particle exited. Reconstructed muon momentum is 2112 MeV. Recorded on 1996/06/09 00:16:29 (run 1938 subrun 21 event 126149).

$\theta = 0$ (not a standard θ convention). The spherical coordinates are translated into polar coordinates with polar ϕ being spherical ϕ and polar r being spherical $\tan\theta$. In this projection r is naturally unbound because $\tan\theta$ diverges for angles approaching 90 degrees. An arbitrary viewing area boundary is imposed. It is a square shape which encloses an area of x and y less than $r_{\text{max}} = \tan(\theta_{\text{open}}/2)$, where θ_{open} is the opening angle of the view.

A small opening angle gives a zoomed view, as through a telescopic lens, similar to an isometric projection. Large θ_{open} gives a fish-eye lens view, which can be quite distorted in extreme cases when θ_{open} approaches 180 degrees. The default opening angle is 100 degrees, chosen to be a little over 90 degrees, in order to be able to see both walls when standing at a corner of the detector.

In this projection straight lines in 3D space remain straight on the screen. This property is very useful in making images very clear when seeing straight rows of PMTs or grid lines outlining the detector. This projection is very easy to understand by people who see it for the first time. Other projections can take some time to get used to. A drawback of this projection is that it does not show a full 4π solid angle view around the viewer. Example of this projection is shown on Fig. 7.4.

7.2.1.8 Stereo Projection

This view is simply a pair of perspective views as seen slightly from the left and from the right of the observer (Fig. 7.5). The distance between the views (between the “eyes”) is adjustable.

The view from the right side of the observer is presented on the left side of the screen and the view from the left side, on the right. To see the stereo effect, user has to look at the right image with the left eye and the left image with the right eye, in effect crossing the eyes as if looking at an object nearby (somewhere between the screen and the eyes), but keeping the eyes focused at the distance of the screen. For most people it takes some practice.

By using a negative distance between the views parameter, the views can be reversed. In such case one views the right image with the right eye and the left image with the left eye. The images on the screen then cannot be too large, and the distance between image centers should not be larger than real distance between the eyes, otherwise the eyes of the observer would have to diverge, which is difficult to do. Observer should imagine looking at an object far away behind the screen but at the same time focusing the eyes at the distance of the screen. This viewing technique is easier to master for some people.

Alternatively the images can be printed and prepared for an easier view, which, however, requires special stereo viewing equipment.

7.2.1.9 Line Discontinuity

When continuous lines in 3D (for example, Cherenkov cones and grids outlining the detector) are projected, discontinuities can occur. For example, in the unrolled view, lines cross from side wall to top or bottom, or in a hemispherical view, lines can cross hemispheres. This condition is easily detected, because real-to-flat functions return wall parameter which is different for each such area and line connecting two points in different areas is not drawn. It is more difficult to detect a discontinuity when a line crosses from one side of the same area to the other side, for example, in the cylindrical projection when it wraps around from left to right or from top to bottom.

A general procedure was developed to detect this condition. As an input, it takes the positions of two points in real space and in screen coordinates. It calculates position in the middle between the two real points and projects it to screen coordinates. Distances between the middle and the end points are calculated in real and window coordinates. Two ratios of the window coordinates divided by the real coordinates are computed. The larger of the two ratios is divided by the smaller one to produce a double ratio. Most of the time this double ratio should be close to one. When the result is greater than 10 it is assumed that the line is discontinuous and the line connecting the two points is not drawn. To save computing time, when the distance between two window pixels is small (less than 4), the continuity test is skipped and the connecting line is drawn.

7.2.2 Color

There are two ways to calculate colors in the color scale: HVC and HSV.

7.2.2.1 HVC

The HVC scale uses the X-windows Color Management System (CMS) to map colors to Tektronics Hue, Value, and Chroma coordinates. This color scale has a complex translation to RGB (Red Green Blue) coordinates. The distance between two points in the HVC coordinate system should be proportional to the percepted difference between colors. Tscan uses maximum Hue and Value settings for any given Chroma.

Since colors in the color scale constructed in this way did not appear uniformly distributed (for example, green region was very small), attempts were made to equalize the scale and to selectively stretch some regions. But this equalization effort was abandoned and a simpler HSV scale was used, which gave much better results.

7.2.2.2 HSV

The Hue Saturation Value color coordinates translate to RGB values in a simple linear way.

A small problem on LCD screens is that red, yellow, green, cyan, blue, magenta colors take a large part of the color scale while colors between them take a smaller part. In other words the red, green and blue components tend to turn on and off almost completely with too little middle values. This was adjusted by a power law formula.

In reverse video mode (light background), some colors, especially around yellow, were too bright with too little contrast with background. To fix this, weights were assigned to color components (blue: 0.8, red: 1.2, green 1.6) and the value (brightness) was truncated if the sum exceeded 1.9.

7.3 Science Popularization

An unexpected benefit of tscan, turned out to be its role in science popularization. Especially the perspective projection, shown in full color on black background, visualizes the beauty of the experiment to untrained eyes.

A muon event, with subsequent decay electron, was shown in a full-page illustration in the cover story of “American Scientist” [94]. “The Particle Odyssey” book [95] includes a colorful full-page event from tscan on the page facing the title page. It also shows side-by-side pictures from tscan that compare showering and non-showering events. A picture of a solar neutrino event, made by tscan, was included in “Sky & Telescope” magazine [96]. A tscan image was a picture of the week on “Particle Physics research in the UK” web site [97]. Images from tscan, with detailed descriptions, are available for the general public at the tscan web pages [92].

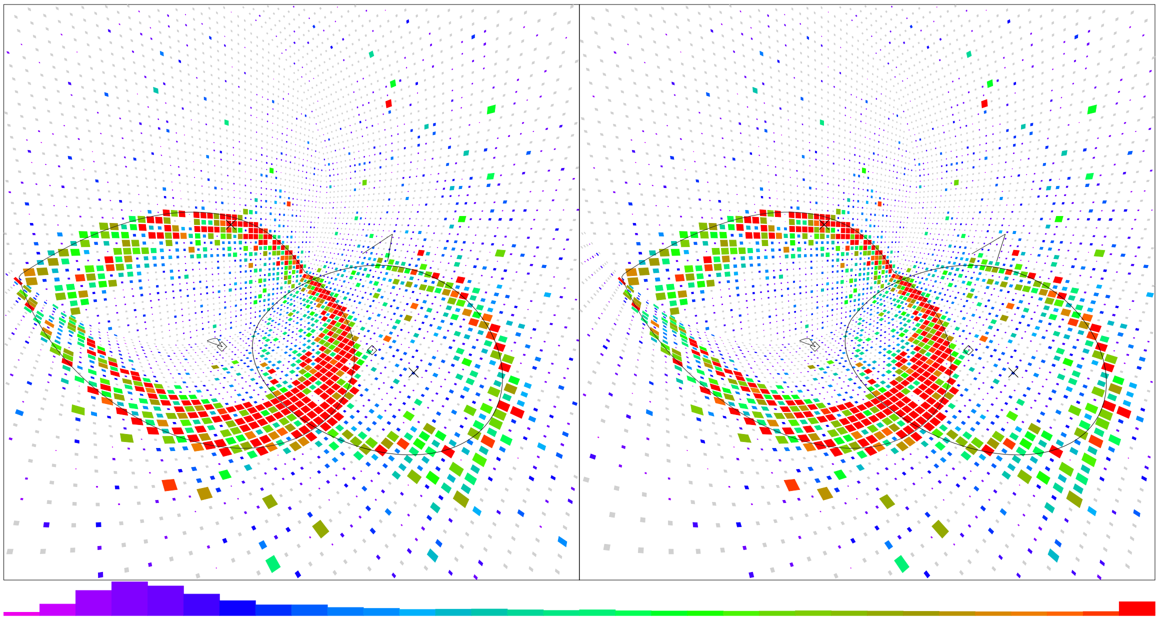


Figure 7.5: A two ring event in stereo view. PMT size and color shows charge. Histogram at the bottom (above a base line shown for color scale) shows number of hits as a function of charge (from 0 to 13 pe). The last bin as an overflow bin. X marks show reconstructed vertex projections on walls. Diamonds mark projected reconstructed track exit points. At the vertex, 2 m long lines are drawn in 3D space in direction of each track (visible above the right ring). Left ring was reconstructed at 1492 MeV, right at 644 MeV. This event was recorded on 1996/06/21 14:08:00 (run 1944 subrun 78 event 474301). The left image is meant for the right eye and the right image for the left eye. Distance between eyes was set to 1 m. To view this event one should cross the eyes. Don't put the image too close. Put your finger tip between the picture and your eyes, close to your eyes. Look at the tip. Put it at such distance that the background SK image blends into 3 images, the middle one is the stereo image. It is blurred as long as you focus on your finger. Now the most difficult part. Try to focus on the SK image without uncrossing your eyes. It may take a few minutes to learn to do it.

Chapter 8

Polfit

Polfit (**P**attern **o**f **L**ight **Fit**ter) is a newly developed particle track fitter for the Super-Kamiokande detector. It works by comparing the observed light (i.e., PMT charge) distribution with the light pattern that would be generated for a given hypothetical configuration of tracks. It varies the track parameters (directions and momenta) until the best match is found. It differs from other track fitters in that it does not look for ring edges, but rather it fits the overall light pattern.

At the heart of the fitter is the likelihood "engine". It can be treated by its users as a black box, taking as input a hypothetical track configuration, and returning a number that tells us how closely the hypothesis matches what is observed in the event that is being processed. On top of the likelihood engine runs a fitting algorithm which picks the initial tracks, decides where the next best candidate hypothesis lies, and, finally, decides when the fit is adequate.

The likelihood engine consists of two major parts. The first part generates the expected light distribution, given a track configuration. The second part compares the expected and the observed light distributions and produces a number which is a function of how similar the two distributions are.

The code for the algorithms described in this chapter can be found in subdirectories (specified below) of the directory `atmpd/src` in the Super-Kamiokande CVS source code repository.

8.1 Expected Light

The expected light (charge) is generated in three stages. First, the direct light is generated, without any absorption or scattering. Then, scattered light is added. Finally, corrections for attenuation and geometry are made, and electronics effects are accounted for to produce the expected charge distribution.

Code for expected light generation is encapsulated in `expq` library located in `expq` directory. This name stands for **exp**ected ***q***, where q is charge. Expected charge distribution is generated by calling the function

```
void
expected_charge(float vertex[3],
                int   nring,
                int   pid[nring],
                float dir[nring][3],
                float mom[nring],
                float angcer[nring],
                float expq[MAXPM],
                float scat[MAXPM]);
```

where `vertex` contains the vertex coordinates, `nring` is the number of tracks (rings), `pid` is an array of particle IDs for each track, `dir` provides direction vectors for each tack, `mom` provides particle momenta for each track, and `angcer` optionally provides Cherenkov angles of each track in radians. The results are put in `expq` array that returns the expected charge for each PMT, and `scat` returns the scattered light. Argument `scat` can be NULL, if `scat_opt.scat_frac` is zero (see description of `scat_opt` on page 118). If `angcer` is NULL, then no correction is done for Cherenkov angle different then theoretical one. Particle ID is in SK convention: 1=gamma, 2=electron, and 3=muon. A special value of `nring` of -99 can be used to generate an isotropic light from the vertex (a point light source). After generation of the expected light, one would ordinarily call `normalize_expected_q()` in order to normalize the expected charge to the observed charge, see Section 8.1.4.

8.1.1 Direct Light

The direct light is generated for each track separately, and then the light for each track is added up.

The Cherenkov photon flux is assumed to be axially symmetric around the track direction (no ϕ dependence). The azimuthal dependence is tabulated as a function of the angle θ, with respect to the track direction.

For very short tacks, viewed from very far away, this parameterization would be sufficient. This would be an approximation of zero-length, point-like tracks viewed

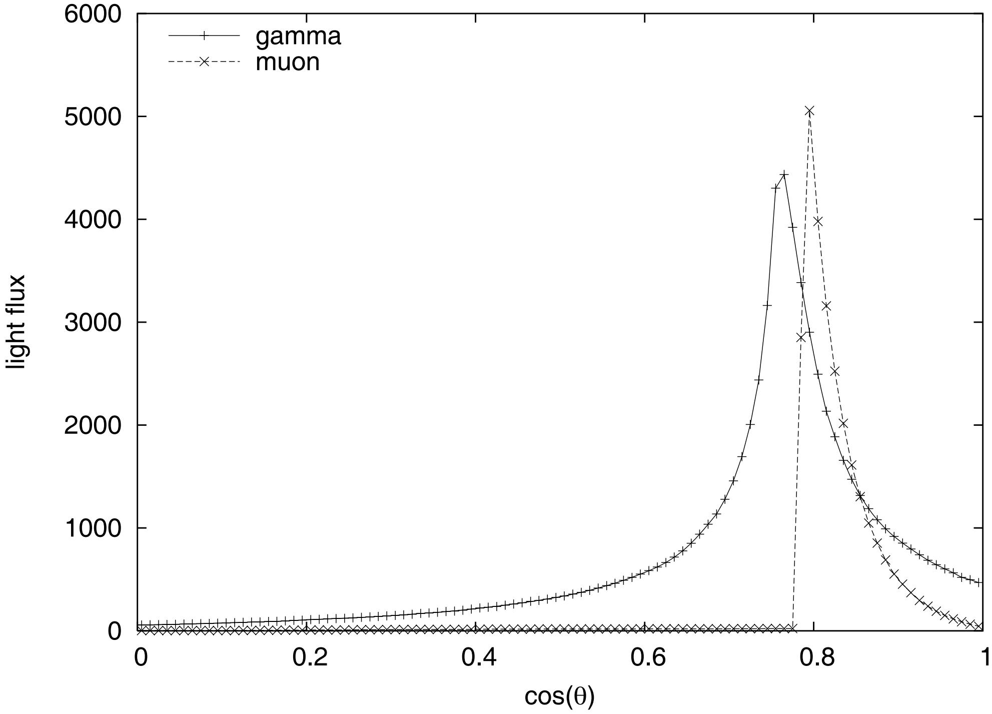


Figure 8.1: Cherenkov light flux of p = 350 MeV gamma and muon at distance of 20 m.

from an infinite distance. However, in the Super-Kamiokande detector, the length of a track (or an electromagnetic shower) and the nearness of the PMTs to the track cannot be ignored. Therefore, the Cherenkov photon flux table depends on another variable, r — the distance of the PMT from the vertex.

Finally the shape of the electromagnetic shower and its length depend on the momentum of the the initial particle — p. This is the third variable of the photon flux table.

There is a separate table for each particle type.

The Cherenkov photon flux tables were generated by a Monte Carlo simulation.

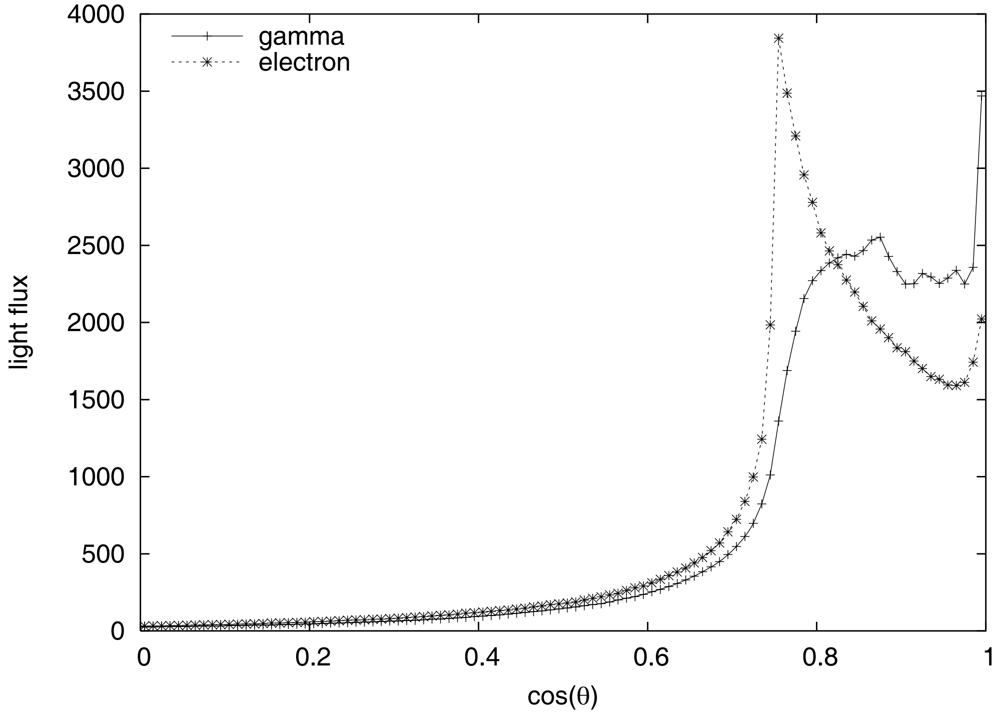


Figure 8.2: Cherenkov light flux of $p = 350$ MeV gamma and electron at distance of 3 m.

For each momentum value in the table, many simulations of mono-energetic particles in water were run. Particle interactions (e.g., development of electromagnetic showers, scattering) were modeled. Charged particles generated Cherenkov photons. The Cherenkov light did not scatter or attenuate in order to generate universal expected light distributions which do not depend on variable water transparency. Absorption and scattering corrections are performed in later steps. The Cherenkov photon flux was measured at various distances and angles. For each distance and angle, the average flux from many simulations was calculated and saved in the table.

For electrons and gammas the flux table has 50 equally spaced values of momentum between 50 and 2000 MeV. There are 30 equally spaced values of r from 100 to 3000 cm. There are 200 equal size $\cos\theta$ bins from -1 to 1.

The photon flux is interpolated (extrapolated) linearly between (outside of) the values of p and r and the centers of $\cos\theta$ bins.

Figure 8.1 shows light flux of gamma and muon, both with momentum of 350 MeV, seen from distance of 20 m. Muon light distribution is sharp, gamma light distribution is diffuse. Also, muon ring opening angle at this momentum is smaller than that of gamma, because of smaller muon velocity (collapsed ring). Figure 8.2 shows light flux of gamma and electron, both with momentum of 350 MeV, seen from distance of 3 m. Amount of light for angles smaller than Cherenkov angle is enhanced due to finite track length, as can be seen by comparing gamma distribution with Figure 8.1. Also, gamma distribution is different from that of electron because gamma travels a finite distance before producing electrons that produce Cherenkov light.

The photon flux tables are contained in files `light_flux_gamma`, `light_flux_electron`, and `light_flux_muon`. These files are in a plain text format. They consist of a header line followed by data lines. All lines consist of space separated fields. The header line consists of the particle ID in the standard PDG Monte Carlo numbering convention [12] followed by three triplets that describe the three dimensions of the parameter space: momentum, distance, and $\cos\theta$. Each triplet consists of the number of bins, the lowest bin value, and the distance between bins. Data lines that follow the header line consist of momentum and distance, followed by flux values for all $\cos\theta$ bins. The data lines come in order of increasing momenta, and within each momentum in order of increasing distance. In other words, the file format is:

```
PID_PDG  p_num p_low p_bin  r_num r_low r_bin  c_num c_low c_bin
p_1  r_1   f_c_1 f_c_2 ... f_c_nc
p_1  r_2   f_c_1 f_c_2 ... f_c_nc
...
p_np r_nr  f_c_1 f_c_2 ... f_c_nc
```

One can register a custom table by calling `register_pid()` with arguments of PID in SK convention, PID in PDG convention, and the file name. If the expected light functions are called without any tables registered, they automatically register the three standard tables for convenience. Calling the function `register_standard_atmpd_pids()` can accomplish the same result. The actual data from the files is read in memory only on the first use of the given particle type.

Function

```
float
expected_light(int pid, float momentum,
               float distance, float cos_theta);
```

returns linearly interpolated Cherenkov light flux for given particle ID `pid`, `momentum`, `distance` from vertex, and $\cos\theta$ `cos_theta`. Internal function `expected_raw_light_flux()`, called from `expected_charge()`, calculates uncorrected direct light distribution on PMTs for given particle track configuration by calling function `expected_light()` for each PMT.

8.1.2 Scattered Light

In the process of the development of the fitter it turned out that direct light is not enough. There are cases when the direct light alone does not adequately describe the observed light distribution. This discrepancy causes incorrect fit results.

Precise calculation of scattered light is very slow. However, the scattered light is a relatively small correction to the overall distribution of the light. A significant amount of time can be saved by calculating it approximately.

Existing code for approximate calculation was too slow. It required a full calculation at every fitter step.

8.1.2.1 Scattering Matrix Method

A new algorithm was developed which requires most of the calculations to be done only once. Only a simple and fast calculation is performed at every fitter step (provided that the vertex does not move).

Calculation of light scattered in water is performed by

```
void
make_scattered_light(float vertex[3],
                     float direct_light[MAXPM],
                     float scattered_light[MAXPM]);
```

where vertex is the event vertex, and direct_light array is the direct light distribution over all PMTs. This function fills scattered_light array with the scattered light distribution.

The most important simplifying assumption in this approach is that all direct light originates at the vertex.

A light ray emitted from the vertex in the direction of any given PMT produces a specific pattern of scattered light on all other PMTs. If we change the intensity of the direct light beam, the intensity of the scattered light pattern, of course, changes linearly, but the overall pattern stays the same. If we knew what scattered light pattern is produced for a direct light emitted in a direction of any PMT, we could simply add the patterns, weighting them with the amount of direct light expected to be seen by each of the PMTs.

In other words, there would be a matrix which tells us how much scattered light a j-th PMT would see for a specific amount of direct light emitted towards an i-th PMT. To get the scattered light distribution we would multiply the vector of the direct light distribution by the scattered light matrix

$$s_j = \sum_{i=1}^{\mathtt{MAXPM}} S_{ij} d_i \ , \tag{8.1}$$

where s_j is the scattered light at the j-th PMT, d_i is the direct light at the i-th PMT, and S_{ij} is an element of the scattering matrix.

However, such a matrix would have $\mathtt{MAXPM}^2 = 11146^2 = 124,233,316$ entries. It would require a large amount of computer memory and manipulating it would be slow.

Fortunately, it turns out, we don't need to consider directions towards every PMT. The scattered light pattern is smooth enough that we can consider many fewer directions.

The detector is divided into patches. For each patch, the direct light seen by all the PMTs which belong to the patch is added. This makes up the direct light vector. After the matrix multiplication we get the scattered light vector. The amount of scattered light for each patch is assigned equally to each of the PMTs in the patch.

The patches used for the direct light and those used for the scattered light don't have to be the same. In fact, we use fewer direct light patches than scattered light patches. The scattering matrix is thus not square in such case. Relatively large direct light patches can be used when the scattering is assumed to be isotropic.

With patches, equation 8.1 becomes

$$s_j = \sum_{i=1}^{\texttt{ndirect}} S_{ij} d_i \ , \tag{8.2}$$

where s_j is the scattered light at the j-th of `nscatter` scattered light patches, d_i is the direct light at the i-th of `ndirect` direct light patches, and S_{ij} is an element of the scattering matrix. There are `ndirect` × `nscatter` matrix elements.

The default patches are shown on Figure 8.3. They are generated by an algorithm which takes a length L parameter as input. The patches are generated with an area as close to L^2 as possible. An attempt is made to give them a compact shape, i.e., not too elongated. For example, on the side wall the ideal shape is square. Neighboring rows of patches are shifted by a half patch width with respect to each other in order to avoid corners with four patches. If $L = 0$, trivial patches are generated, i.e., each patch consists of a single PMT. For the direct light patches $L = 800$ cm. For the

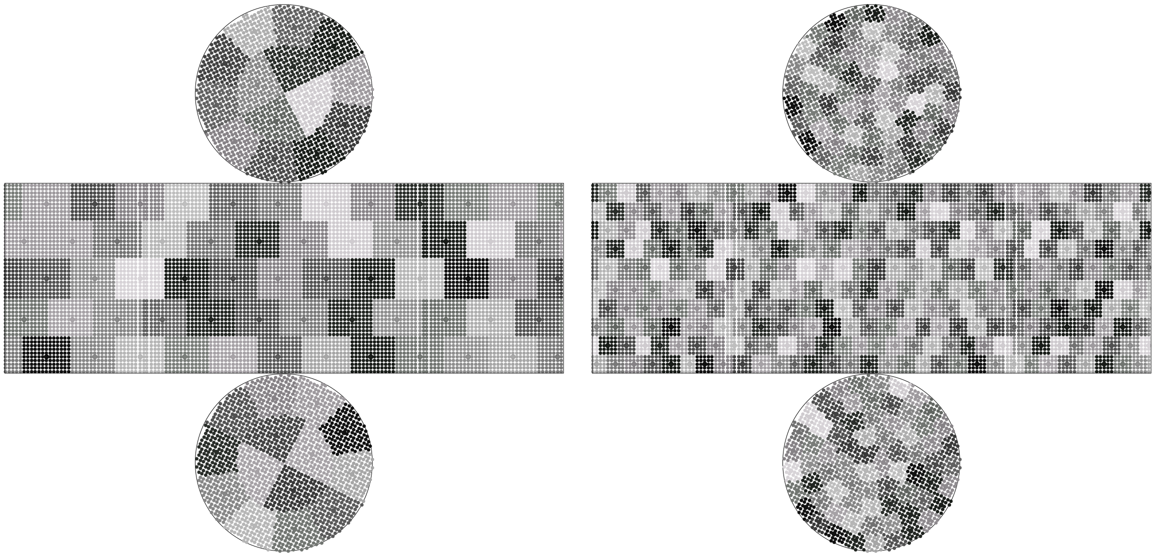


Figure 8.3: The patches used for the scattered light calculation. Each patch is assigned a random shade of grey. Central PMT of each patch is enlarged. The direct light patches are shown on the left, the scattered light patches are shown on the right. This figure was made with tscan, using a generated synthetic event in text format (see Section 7.1.2) with PMT time determined by the patch to which it belongs (each patch having a random time), and charge of central PMTs larger than remaining PMTs.

scattered light patches $L = 5 \times 71$ cm (71 cm is the distance between neighboring PMTs), i.e., the ideal patch size is $5^2 = 25$ PMTs.

The scattering matrix is calculated once before the fitting iterations.

8.1.2.2 Patch Generation and Manipulation

Patches are calculated by the function `init_scattered_light()`, normally called automatically the first time scattered light calculation is performed. Allocated memory can be freed by calling `end_scattered_light()`. Functions that calculate and manipulate patches are general-purpose and can be used independently of the rest of this framework.

The function:

```
int
make_patches(float size, int pmt_patch_idx[MAXPM]);
```

calculates the detector patches of given linear `size` (L). Each phototube in `pmt_patch_idx` array is assigned a patch index number. This function returns the total number of created patches (`npatch`). This number should be used to allocate the remaining patch structures. If `size` is zero, trivial patches are created, with one PMT per patch.

This function first calculates the number of bins on the side wall in the vertical z direction:

$$b_z = \frac{H}{L} \, , \tag{8.3}$$

where H is the height of the detector. The number of bins is always rounded to the nearest integer, and if the result is 0 it is set to 1. Next, corrected patch size is calculated, in order to get the area of the patches on the side wall closer to the desired area, by reducing the chance that both dimensions are rounded in the same direction (both up, or both down):

$$L_\phi = \frac{L^2}{H/b_z} \, . \tag{8.4}$$

Then, the number of ϕ bins on the side wall is calculated

$$b_\phi = \frac{\pi D}{L_\phi} \, , \tag{8.5}$$

where D is the diameter of the detector. Next, intermediate number of diameter bins on the top and bottom walls is calculated

$$b_d = \frac{D}{L} \ , \tag{8.6}$$

and, then, the number of radial bins is calculated:

$$b_r = \frac{b_d + 1}{2} \ . \tag{8.7}$$

Next, for each radial bin i, we calculate its outer radius (r_i) and number of its bins ($b_{\phi i}$). For the the first bin (the inner circle), its temporary radius is

$$r'_1 = \frac{D}{b_r} \ . \tag{8.8}$$

If the number of diameter bins b_d is odd, we halve r'_1. The remaining radial bins (rings) have temporary outer radius

$$r'_i = r_{i-1} + \frac{D/2 - R_{i-1}}{b_r - i + 1} \ . \tag{8.9}$$

Next, temporary area of the i-th ring is calculated:

$$A'_i = \pi r'^2_i - \pi r^2_{i-1} \ . \tag{8.10}$$

The number of ϕ bins in the i-th ring is

$$b_{\phi i} = \frac{A'_i}{L^2} \ . \tag{8.11}$$

Next, we calculate the ideal outer bin radius in order to have the patch area closer to the desired area (we don't correct the outer radius of the last bin—it is determined by the radius of the detector):

$$r_i = \sqrt{\frac{b_{\phi i} L^2}{\pi} + r_{i-1}^2}\,. \tag{8.12}$$

Once the height, circumference, radial, and ϕ bin sizes are determined, each PMT is assigned to a patch, by determining within which patch the center of the PMT resides. Before that, every second row of patches along height and radius directions is shifted along ϕ direction by half of patch size in order to avoid corners with four patches, and to get a more uniform distribution of patch centers.

Function

```
void
count_patch_pmts(int npatch,
                 int pmt_patch_idx[MAXPM],
                 int patch_counts[npatch]);
```

takes as input the total number of patches `npatch`, and `pmt_patch_idx` which is the array filled by `make_patches()`. It fills each i-th element of the `patch_counts` array with the number of PMTs (N_i) that are in the set of PMTs (p_i) that belongs the i-th patch.

Function

```
void
patch_central_pmt(int npatch,
                  int pmt_patch_idx[MAXPM],
                  int central_pmts[npatch]);
```

takes as input the total number of patches `npatch`, and `pmt_patch_idx` which is the array filled by `make_patches()`. It returns in the `central_pmts` array indices of

PMTs that are closest to the center of their patches. This function works by, first, calling an internal function find_patch_centers() in order to find the center of mass of PMTs that belong to each patch,

$$\boldsymbol{M}_i = \frac{1}{N_i} \sum_{j \in p_i} \boldsymbol{P}_j \ , \tag{8.13}$$

where $\boldsymbol{P}_j$ is the position of j-th PMT. It then calculates the distance to the center of mass of each PMT that belongs to the patch,

$$d_{ij} = |\boldsymbol{P}_j - \boldsymbol{M}_i| \ , \ \ j \in p_i \ , \tag{8.14}$$

and then puts into i-th element of the central_pmts array the index of the PMT that is closest to the center of mass of i-th patch.

The above three functions are called from function init_scattered_light().

Function

```
void
patch2pmt_charge(int   pmt_patch_idx[MAXPM],
                 float patch_values[npatch],
                 float pmt_values[MAXPM]);
```

takes as input pmt_patch_idx prepared by make_patches(), and an array of values (for example, charges) assigned to each patch, patch_values, and adds this value for each PMT in the patch in the pmt_values array. This function is used by function make_scattered_light(), as described on page 118.

8.1.2.3 Scattering Matrix Elements

A light ray leading from the vertex to the central PMT of a direct light patch is considered. This light ray is a linear source of secondary light, i.e., scattered light. Due to attenuation, each segment of the linear source gets dimmer exponentially further away from the vertex.

The source is viewed from the central PMT of a scattered light patch. To calculate the amount of the scattered light from the light ray reaching the PMT, integration is performed along the linear source. The integrating algorithm samples the amount of the scattered light emitted towards the PMT from several discrete points along the linear source. The linear source is split into segments (one for each point) and each point represents the total amount of the scattered light from its segment. By default, Romberg integration is used. In such case, the line is divided into 9 equal segments with a point in the middle of each segment. A faster, but less accurate, 3 point integration is also available.

At each point, the amount of the scattered light is the total amount of the light attenuated in the corresponding line segment.

The total amount of the light to be scattered is divided into two parts: an isotropic part and an angle-dependent part. The angle-dependent part represents the Rayleigh scattering and it scatters with an angular distribution of $\cos^2(\theta)$, i.e., it is a forward-back scattering. By default, the angle-dependent fraction is set to zero, i.e., only the isotropic scattering is calculated.

The intensity of the scattered light seen from any point falls off as $1/R^2$. However, this formula would give very large intensities for small distances. The point represents a diffuse light source which occupies a volume of a segment (1/9-th) of the solid angle of the direct light patch as seen from the vertex. To avoid the unwanted divergence, R is restricted to values greater than $R_{\min}$. By default, $R_{\min} = 400$ cm, which is comparable with the size of a direct light patch.

The primary light beam from the vertex to the central PMT of the direct light patch represents a photon flux from the vertex to the whole patch. The flux is proportional to the solid angle of the patch as viewed from the vertex. Therefore, the intensity of the light beam is adjusted, taking into account the distance to the patch, and the angle of the patch plane with respect to the light beam.

8.1.2.4 Scattering Matrix Calculation

Function make_scattering_matrix() prepares the scattering matrix for a given vertex. It is called automatically from make_scattered_light() (see page 103) each time new vertex is used. Function make_scattering_matrix() calls

```
void
fill_scatt_matrix(float vertex[3],
                  float scat_frac,
                  float scat_len,
                  float scat_min_r,
                  bool  scat_9pt,
                  float scat_ang_frac,
                  bool  attenuate_scattered,
                  int   ndirect,
                  int   direct_cent[ndirect],
                        int   direct_count[ndirect],
                        int   nscatter,
                        int   scatter_cent[nscatter],
                        float scatt_matrix[ndirect][nscatter]);
```

that does the main work. Argument `vertex` is the event vertex $\boldsymbol{V}$, `scat_len` is the scattering length, `scat_frac`, `scat_min_r`, `scat_9pt`, and `scat_ang_frac` have the same meaning as in the `scat_opt` structure described on page 118. `attenuate_scattered` is a flag to turn on attenuation of scattered light. `ndirect` is the number of direct light patches, `direct_cent` array contains the indices of PMTs that are at the centers of direct light patches, `direct_count` array contains the number of PMTs in each direct light patch, `nscatter` is the number of scattered light patches, and `scatter_cent` array contains the indices of PMTs that are at the centers of scattered light patches. The resulting scattering light matrix is returned in `scatt_matrix` array. Function `make_scattering_matrix()` can optionally call `fill_scatt_matrix()` second time, in order to add short wavelength scattering based on parameters in `scat_opt` structure described on page 118 .

Function `fill_scatt_matrix()`, first, iterates over all direct light patches. For each direct light patch, with the central PMT coordinates $\boldsymbol{D}$, it calculates the vector of the direct light from the vertex to the central PMT

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{V} \ , \tag{8.15}$$

and the unit direction vector from the vertex to the central PMT

$$\boldsymbol{l} = \frac{\boldsymbol{L}}{|\boldsymbol{L}|} \ . \tag{8.16}$$

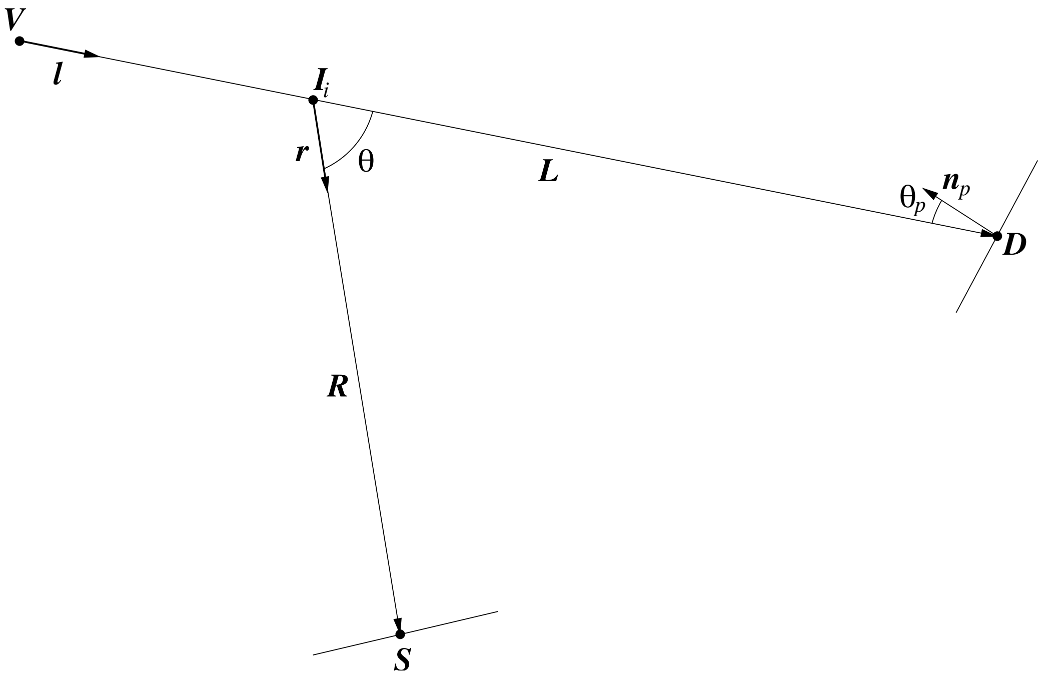


Figure 8.4: Schematic view of geometry of scattering matrix element calculation. $\boldsymbol{V}$ is the vertex, $\boldsymbol{D}$ is the center of the direct light patch, $\boldsymbol{I}_i$ is the i-th integration point, and $\boldsymbol{S}$ is the center of the scattered light patch.

It then calculates corrections needed to transform the uncorrected light flux to the actual light flux reaching the direct light patch. The first correction takes into account the viewing angle of the patch with respect to the vertex

$$c_n = \cos\theta_p = -\boldsymbol{l} \cdot \boldsymbol{n}_p \;, \tag{8.17}$$

where θ_p is the angle of the central PMT with respect to the vertex, and $\boldsymbol{n}_p$ is the normal direction of the central PMT. The minus sign is needed because, by convention, $\boldsymbol{n}_p$ points into the inside of the detector, while $\boldsymbol{l}$ points outside. Next correction is for the solid angle of the patch, as viewed from the vertex. The patch is approximated by a circle with radius

$$r_d = \frac{n_d d_p^2}{\pi} \ , \tag{8.18}$$

where n_d is the number of PMTs in the direct light patch, and d_p is the distance between neighboring PMTs, i.e., 71 cm. The solid angle correction is then

$$c_a = 1 - \frac{|\boldsymbol{L}|}{\sqrt{|\boldsymbol{L}|^2 + r_d^2}} \ . \tag{8.19}$$

This is analogous to equation 8.30. The total combined correction is

$$c_f = c_n c_a \frac{1}{n_d} \ . \tag{8.20}$$

Factor of $1/n_d$ is used, because the charge assigned to the direct light patch is the total charge observed by constituent PMTs, not the average charge.

Next, internal function `direct_attenuation()` is called in order to calculate the amount of light scattered at each $\boldsymbol{I}_i$ point of N integration points along the direct light ray $\boldsymbol{L}$, at distance l_i from the vertex. First, a segment of the light ray is assigned to each integration point, with segment boundaries half-way between each integration point

$$l'_i = \frac{l_i + l_{i+1}}{2} \tag{8.21}$$

for $1 \leq i \leq N - 1$, with $l'_0 = 0$, and $l'_N = l_N$. The relative intensity of direct light at each segment boundary is

$$i_i = e^{-l'_i/\lambda} \ , \tag{8.22}$$

where λ is the attenuation length passed as parameter `scat_len`. The amount of scattered light in the segment for i-th integration point $\boldsymbol{I}_i$ is

$$a_i = i_i - i_{i-1} \ . \tag{8.23}$$

After this, `fill_scatt_matrix()`, for each of direct light patches, iterates over scattered light patches, and performs integration. Integration is performed on range from 0 to 1, where 0 corresponds to the vertex, 1 corresponds to the central PMT of the direct light patch, and values in between correspond to the points on the light ray connecting these two points. One of two integration methods is used, based on `scat_9pt` parameter.

A simple three point integration uses three integration points $x_i : (0, 0.5, 1)$, i.e., (vertex, mid-point, direct light patch). The integral is calculated as follows:

$$\int_0^1 f(x)dx = \frac{\sum_i w(x_i)f(x_i)}{\sum_i w(x_i)} \ , \tag{8.24}$$

where $w(x_i)$ are the weights. The code currently uses weights $w(x_i) : (1, 2, 1)$, i.e., trapezoid integration is performed, based on a first order polynomial. Weights $w(x_i) : (1, 4, 1)$ give Simpson's integration, based on a second order polynomial. Other weights are possible too.

In order to perform 9-point Romberg integration, function `rromo1()` is called.

During integration, an internal function `scatter()` is called that calculates the amount of scattered light from a given integration point $\boldsymbol{I}_i$ to the scattered light patch. This function is relatively time critical because it is called for each integration point for each scattered light patch for each direct light patch. However, it turns out, that total time added to fitting is negligible, because this calculation is performed only once for each fit. First, it calculates the vector from the integration point to the position $\boldsymbol{S}$ of the central PMT of scattered light patch

$$\boldsymbol{R} = \boldsymbol{S} - \boldsymbol{I}_i \ , \quad R_k = S_k - I_k \ , \tag{8.25}$$

where $\boldsymbol{L}$ is given by equation 8.15, and x_i is the argument used in equation 8.24. It also calculates the squared distance between these two points

$$R_{\text{sq}} \equiv R^2 \equiv |\boldsymbol{R}|^2 = \boldsymbol{R} \cdot \boldsymbol{R} = \sum_{k=1}^{3} R_k^2 \ . \tag{8.26}$$

Then, if `scat_ang_frac` is nonzero, function `scatter()` calculates the angle dependent part of scattered light

$$c_s \equiv 2\cos^2\theta = 2(\boldsymbol{l} \cdot \boldsymbol{r})^2 = 2\frac{\left(\sum_k l_k R_k\right)^2}{R_{\text{sq}}} \ , \tag{8.27}$$

where θ is the angle between direction of direct light and direction of scattered light, $\boldsymbol{l}$ is defined in equation 8.16 and $\boldsymbol{r} = \boldsymbol{R}/R$. Factor of 2 is used in order to get the same normalization as for isotropic scattering. Next, if `attenuate_scattered` is set, function `scatter()` calculates attenuation factor

$$a_s = e^{-R/\lambda} \ , \tag{8.28}$$

where $R = \sqrt{R_{\text{sq}}}$, and λ is parameter `scat_len`. Note, that this operation requires relatively slow calculations of square root and an exponent. Finally, the result for this integration point is calculated as

$$a_i a_s (f_i + f_a c_s) \frac{1}{R'_{\text{sq}}} \ , \tag{8.29}$$

where a_i is given by equation 8.23, $f_i = 1 - f_a$ is the isotropic scattering fraction, f_a is the angle-dependent scattering fraction given by parameter `scat_ang_frac`, $R'_{\text{sq}} = R_{\text{sq}}$ if $R_{\text{sq}} > R^2_{\text{min}}$ or $R'_{\text{sq}} = R^2_{\text{min}}$ otherwise, and R_{min} is the `scat_min_r` parameter.

After the scattering matrix is calculated, function `make_scattered_light()` (see page 103) calculates direct light reaching each of the direct patches, multiplies this by the scattering matrix, using equation 8.2, in order to get values in the scattered light patches, and then assigns values from scattered light patches to all individual PMTs by calling the function `patch2pmt_charge()` (see page 110).

Parameters that control scattered light calculation, when calling `make_scattered_light()` function, are stored in `scat_opt` structure. This structure contains the following fields:

- `scat_frac` sets the intensity of scattered light. 1 is normal intensity, 0 turns off scattered light, values below 1 decrease scattered light, values above 1 increase scattered light. Default is 1. Scattered light is not attenuated by default. When this value is nonzero, the size of direct light patches (field `patch_size_direct` below) should be decreased.

- `scat_len` is the scattering length in cm. Value of -1 (the default) causes the length to be taken from the standard `skwaterlen` parameter.

- `scat_min_r` is the minimum distance $R_{\min}$ for $1/R^2$ correction in cm in equation 8.29, by default, 400.

- `scat_9pt` is a flag that, when set, chooses nine point integration, and, when cleared, selects three point integration. Default is set.

- `scat_ang_frac` controls the amount of anisotropic $\cos^2(\theta)$ scattering f_a in equation 8.29. 0 (the default) turns it off (i.e., all scattering is isotropic), 1 causes all scattered light to be of this kind, and values in between choose mixture if anisotropic and isotropic scattering.

- `swl_frac` controls the amount of the short wavelength scattering. This scattered light is attenuated. 0 (the default) turns it off, 1 causes all scattered light to be of this kind, and values between 0 and 1 select mixture of scattered lights.

- `swl_len` is the scattering length of short wavelength scattering in cm. Typical values would be much shorter than ordinary scattering length. Default is 0.

- `patch_size_direct` is the linear size of direct light patches in cm. Default value is 800.

- `patch_size_scatter` is the linear size of scattered light patches in cm. Default value is 355, i.e., 5×71.

8.1.3 Corrections

The scattered light was calculated based on a direct light photon flux. After that, the detector effects are added to the direct light. A series of corrections is applied which translate the photon flux into charge observed by the PMTs.

The largest correction is due to the solid angle effect. A PMT close to the vertex sees much more Cherenkov photons than a PMT far away because it covers a larger solid angle as seen from the vertex. The direct light is multiplied by a factor

$$1 - \frac{r}{\sqrt{r^2 + r_{\mathrm{PMT}}^2}} \,, \tag{8.30}$$

where r is the distance of the PMT from the vertex, and r_{PMT} is the PMT radius. This is analogous to equation 8.19. Internal function `correct_solid_angle()` performs this calculation.

Another correction is for PMT acceptance variation with the incident angle of the light, performed by internal function `correct_acceptance()`, that calls `coseffsk(`$\cos\theta$`)`, where θ is the angle between the direction of the light from the vertex and normal direction of the PMT. Argument $\cos\theta$ is calculated using method analogous to equation 8.17.

The final correction is for water attenuation. Different wavelengths attenuate differently in water but a single exponential describes the effective attenuation well enough. By default, the attenuation length used for the calculation is the measured effective attenuation length in the Super-Kamiokande detector at the time when the fitted data was taken, i.e., the value is taken from the standard `skwaterlen` parameter. In the case of the Monte Carlo, the same attenuation length as the one used in the Monte Carlo simulation is used, also taken from the `skwaterlen` parameter. The expected charge is multiplied by a factor $e^{-r/\lambda}$, where λ is the attenuation length. Internal function `correct_attenuation()` performs this calculation. If short wavelength scattering is enabled in `scat_opt` (see above) then function `correct_attenuation_2_scales()` is used instead, in order to use double exponential attenuation.

In order to speed-up the calculation, an array of correcting factors, one for each PMT, is calculated only once for each vertex, and it is cached between invocations for the same vertex. The direct light for each PMT is simply multiplied by the correcting factor for that PMT.

After the corrections are performed, the scattered light is added to the direct light.

8.1.4 Normalization

The final step of the expected light calculation is the normalization. The expected light is normalized by the observed light.

Only observed PMT hits within a specific timing window are counted (see Section 8.3.2 on page 129). The remaining PMTs are treated as if they were not hit by any photon. By default, PMTs with an observed charge below 1 pe (photoelectron) are assigned a single pe value. This is done because the PMT which recorded a hit observed at least one photon. Bad PMTs are ignored in both the expected charge and the observed charge arrays. After these adjustments, all the observed PMT charge is added up. The expected charge is multiplied by a factor which makes the total expected charge identical to the total observed charge.

Real PMTs saturate at a high charge. This has to be accounted for consistently in the normalization calculation and later in the likelihood calculation. A maximum charge value is used, by default, 200 pe, which is somewhat below the typical saturation levels. Both the expected and the observed charge sums are recalculated again but with all charges above the maximum charge counted as the maximum charge. The expected charge is renormalized again to the “saturated” observed charge using the truncated sums. This procedure is repeated until the normalization converges.

Normalization is performed by function

```
float
normalize_expected_q(float expected[MAXPM],
                     float observed[MAXPM],
                     int   bad[MAXPM]);
```

where `expected` is array of the expected charge, `observed` is array of the observed charge, and `bad` is the array of bad PMTs. After this function returns, `observed` array is normalized. However, it is not truncated at saturation point. The return value is the overall normalization factor for the expected light.

8.2 Likelihood

At this point we have a normalized expected charge distribution. We can compare it to the observed charge distribution. In the comparison bad PMTs are ignored. The observed charge has a timing window cut applied (see Section 8.3.2 on page 129).

One of two comparison methods is used.

The first method calculates for each PMT a likelihood l_{PMT} that, for a given amount of the expected charge, the PMT records the observed amount of charge. Standard function `spprobpe(peexp, peobs)` calculates l_{PMT}. It uses lookup tables for expected charge below 10 pe and Gaussian distribution above. A logarithm of the likelihood is taken and the log-likelihoods of all the PMTs are added up (an operation equivalent to multiplying all the likelihoods)

$$l_l = \sum_{\mathrm{PMT}} \ln l_{\mathrm{PMT}} \ . \tag{8.31}$$

Every individual likelihood is constrained to be above a minimum value, by default, the minimum is 10^{-7}. The sum is the overall log-likelihood that tells us how well the observed and the expected light distributions match. The number is always negative. The larger the number (less negative), the better is the match.

The second comparison method is the one used by default. A simple Poisson χ^2 formula is used [98]:

$$\chi^2_{\mathrm{PMT}} = 2(q_e - q_o) + 2q_o \ln \frac{q_o}{q_e} \ , \tag{8.32}$$

where q_e is the expected charge, and q_o is the observed charge. This simple formula turns out to give as good fit results as the likelihood method. By default, before the calculation, all the observed charge below 1 pe is set to 1 pe level for consistency with the normalization calculation. Also, by default, when both the observed and the expected charges are above the saturation level used in the normalization calculation (see Section 8.1.4), the expected charge is set to the saturation level. The values of χ^2_{PMT} for every PMT are added:

$$\chi^2 = \sum_{\mathrm{PMT}} \chi^2_{\mathrm{PMT}} \, . \tag{8.33}$$

The total sum of the overall χ^2 tells us how well the observed and the expected light distributions match. The number is always positive. The smaller the number, the better is the match. For each PMT, χ^2_{PMT} is calculated by function

```
float
get_chi2(float pe_expected, float pe_observed,
         float min_pe_obs,  float pe_satur);
```

where `pe_expected` is the expected charge, `pe_observed` is the observed charge, `min_pe_obs` is the single photoelectron level, and `pe_satur` is the saturation level.

Function

```
float
likelihood_from_exp_q(float pmt_exp[MAXPM],
                      float pmt_obs[MAXPM],
                      int   pmt_bad[MAXPM],
                      float pmt_lik[MAXPM]);
```

calculates total log-likelihood l_l if `lik_opt.old_likelihood` is set (nonzero), or it calculates $-\chi^2$ if `lik_opt.old_likelihood` is cleared (this is the default). `pmt_exp` array is the expected charge distribution over all PMTs, `pmt_obs` array is the observed charge distribution, and `pmt_bad` array lists bad PMTs. Log-likelihood (or $-\chi^2$) contribution of each PMT is returned in `pmt_lik` array, and the function returns total log-likelihood (or $-\chi^2$).

The main function for the likelihood engine, that calculates likelihood based on a hypothetical event configuration (vertex and tracks), is

```
float
track_likelihood(float pmt_obs[MAXPM],
                 int   pmt_bad[MAXPM],
                 float vertex[3],
                 int   nring,
                 int   pid[nring],
                 float dir[nring][3],
                 float mom[nring]);
```

where `pmt_obs` is the array of observed PMT charges, `pmt_bad` array specifies which PMTs are bad, and `vertex`, `nring`, `pid`, `dir`, and `mom` specify the event configuration: vertex, number of rings, particle ID of each ring in SK convention, track directions, and track momenta. This function returns the log-likelihood or $-\chi^2$. This function calls `expected_charge()` (see Section 8.1, page 97) and `normalize_expected_q()` (see Section 8.1.4), and then it calls `likelihood_from_exp_q()`.

Code for likelihood calculations, and fitting is contained in `polfit` directory.

8.3 Fitter

Using the general likelihood engine described above fitters can be developed which search for the configuration of rings which best matches the observed pattern of light.

A specialized π^0 fitter was developed. It is designed to run on a sample of events which were reconstructed as a single showering ring. It tries to reconstruct two rings, each created by a gamma ray from the decay

$$\pi^0 \to \gamma\gamma \; . \tag{8.34}$$

As input, it takes the vertex, the total energy of both gamma rays and the direction of the first gamma ray. Typically the input parameters are taken from the reconstructed vertex, the visible energy and the direction of the reconstructed ring, respectively. It fits the direction of the second gamma ray and the fraction of the total energy taken by the second gamma.

8.3.1 Initial Guess—Inertia Tensor Eigenvectors

Before the fitting starts, an estimate is made of where the plane, in which the two gamma rays lie, is located.

First, timing and clustering cuts are performed on the observed charge using function `riselpmt()`. The observed charge is then corrected for attenuation and PMT acceptance using `ripecorr()`. Function `correct_pmt()`, called from `get_tensor()` (see below) performs these tasks.

The charge is then projected onto a sphere centered around the vertex. This is done by calculating position of each PMT relative to the vertex,

$$\boldsymbol{R}_i = \boldsymbol{P}_i - \boldsymbol{V} \ , \tag{8.35}$$

where $\boldsymbol{V}$ is the vertex, and $\boldsymbol{P}_i$ is the position of the i-th PMT. Then, the coordinates on the sphere of radius 1 m are calculated:

$$\boldsymbol{r}_i = \frac{\boldsymbol{R}_i}{|\boldsymbol{R}|} \ . \tag{8.36}$$

The projected charge is treated as a mass distribution and an inertia tensor is calculated by calling

```
void
inertia_tensor(int   nmass,
               float mass[nmass],
               float pos[nmass][3],
               float inertia[3][3]);
```

where `nmass` contains number (N) of PMT hits that represent masses, `mass` array contains the charge of each PMT that represents amount of mass m_i, and `pos` array contains projected PMT positions $\boldsymbol{r}_i$ that represent positions of each mass. This function returns the inertia tensor in the `inertia` array. The inertia tensor is

$$I_{jk} = \sum_{i=1}^{N} m_i (r_i^2 \delta_{jk} - r_{i,j} r_{i,k}) \ . \tag{8.37}$$

Because I_{jk} is symmetric, `inertia_tensor()` function calculates only the upper half (and the diagonal), and copies the upper half to the lower half. For $j \neq k$, the result is

$$I_{jk} = I_{kj} = -\sum_{i=1}^{N} m_i r_{i,j} r_{i,k} \ , \quad j \neq k \ , \tag{8.38}$$

and for $j = k$, the result is

$$I_{jj} = \sum_{i=1}^{N} m_i (r_{\mathrm{sq}\,i} - r_{i,j}^2) \ , \tag{8.39}$$

where

$$r_{\mathrm{sq}\,i} \equiv r_i^2 = \sum_{k=1}^{3} r_{i,k}^2 \ . \tag{8.40}$$

The tensor is diagonalized and the eigenvectors with the corresponding eigenvalues are obtained. This is performed using function `ssyev()` from LAPACK (Linear Algebra PACKage) [99].

The eigenvector corresponding to the smallest eigenvalue determines the direction along which most of the charge is located. The eigenvector corresponding to the largest eigenvalue determines the direction perpendicular to the plane in which most of the charge is located. In an ideal case, with perfect Cherenkov rings projected onto a sphere, this plane would be the $\gamma - \gamma$ plane. Detector effects and fluctuations cause deviations from the ideal case.

The fitter calculates the inertia tensor by calling

```
void
get_tensor(float vertex[3], float main_dir[3],
           float eigval[3], float eigvec[3][3]);
```

where `vertex` is the event vertex, `main_dir` is the main event direction, `eigval` returns the eigenvalues, and `eigvec` returns the eigenvectors. `main_dir` is used to chose the sign of the first two eigenvectors to point towards that direction, and not away from it, by possibly reversing them. It ensures that the eigenvectors don't point away from the main track. This direction can be derived, for example, from the direction in which the majority of charge is present. In π^0 fitter, we use the direction of the first reconstructed track. Function `get_tensor()` gets the PMT hit information from the standard SK common blocks.

Code for the inertia tensor calculation `inertia_tensor()` is located in in `ska` directory. This is the same code as the one used by tscan, see Section 7.2, page 80.

8.3.2 Iterations—Simplex

The fitter varies three variables: two angles, ϕ and θ, which parameterize the direction of the second ring in a spherical coordinate system, and a fraction f of the total energy carried by the second gamma. The spherical coordinate system is defined in such a way that the first gamma lies on the equator with both ϕ and θ equal to zero. The poles at $\theta = \pm 90^o$, the first ring, and the eigenvector corresponding to the largest eigenvalue lie in one plane. In this way the equator plane approximates the $\gamma - \gamma$ plane determined by the inertia tensor, given the constraint that the equator contains the first ring. Negative ϕ direction is chosen on the same side as the eigenvector corresponding to the smallest eigenvalue. This is the side of the first ring on which the majority of the charge is located and where we expect to find the second ring.

Before the fitting proceeds, a timing cut is performed on the observed PMT hits by function `timing_cut()` that in turn calls `ritofcut()`. First, the time of flight of light from the vertex to every PMT is subtracted. A sliding timing window of a constant width is moved until the maximum number of hits is found. The width of the window is 30 ns for events with a total number of pe corresponding to an energy below about 1 GeV and it is expanded linearly for events with more charge.

The fitting is done using a simplex method. For an N-parameter fit in an N-dimensional parameter space a set of $N+1$ points $\boldsymbol{V}_i$ (called simplex) is considered at every fitting step. The simplex points cannot lie on an $N-1$-dimensional hyperplane, i.e., the (hyper)volume contained in a boundary with corners defined by the simplex points must be non-zero.

The initial simplex (the set of points) must also fulfill this condition. The simplest way to ensure this is to pick one point, and the remaining N points are set by moving the first point along one coordinate, different one for each point.

$$V_{i+1,k} = V_{1,k} + D_i \, \delta_{ik} \ , \tag{8.41}$$

where $\boldsymbol{D}$ contains the shift coordinates.

In our case, $N = 3$. By default, the first point is picked at the coordinates $\boldsymbol{V}_1(\phi, \theta, f) = (-30, 0, 0.4)$ (the angles are measured in degrees). The remaining points are chosen by changing one coordinate to one of the values $\boldsymbol{D} = (20, 20, 0.2)$.

The fitting stops when the distance between the best (b) and the worst (w) simplex points is less than 1, where the distance is measured using units for each axis defined by the metric $\boldsymbol{M} = (2, 2, 0.02)$:

$$\sum_{k=1}^{N} \left(\frac{V_{b,k} - V_{w,k}}{M_k} \right)^2 < 1 \ . \tag{8.42}$$

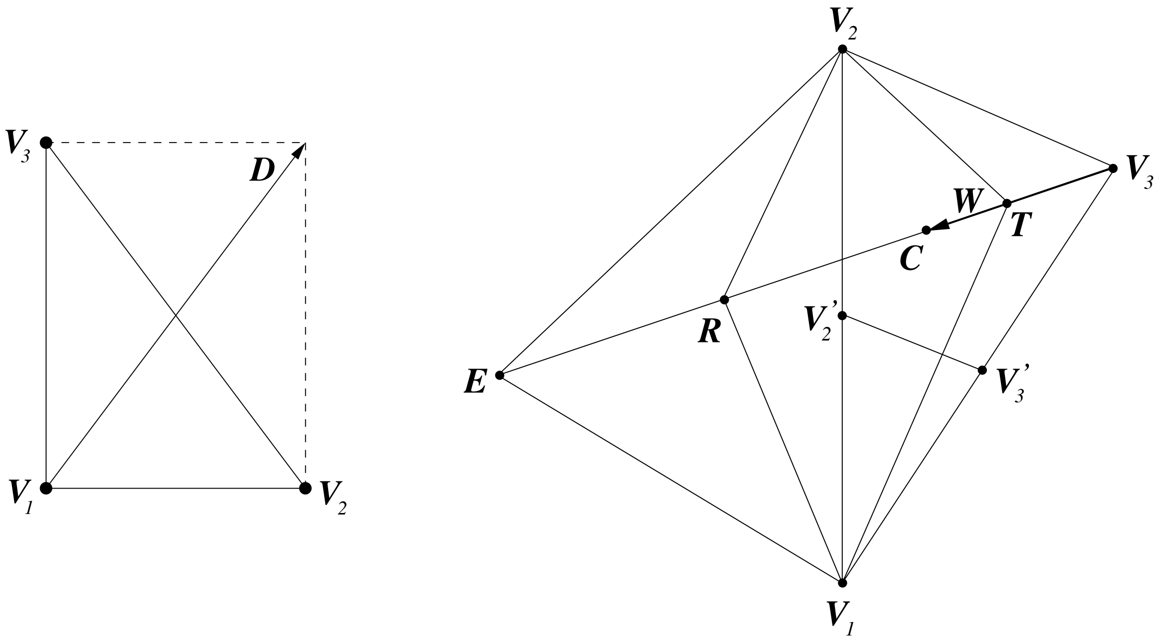


Figure 8.5: Schematic view of two-dimensional ($N = 2$) simplex calculations. Left-hand side shows the initial simplex calculation based on the initial point $\boldsymbol{V}_1$ and the coordinate displacement vector $\boldsymbol{D}$. Right-hand side shows points that could be considered during an iteration step. In this example, $\boldsymbol{V}_1$ is the best point, and $\boldsymbol{V}_3$ is the worst point.

At each iteration point, a centroid of the simplex is calculated

$$\boldsymbol{C} = \frac{1}{N+1} \sum_{i=1}^{N+1} \boldsymbol{V}_i \ . \tag{8.43}$$

The worst point $\boldsymbol{V}_w$ may be left out from this calculation, if field `flat_center` of structure `simplex_opt` is set, see page 135. Subsequent operations are performed with respect to the centroid $\boldsymbol{C}$. Vector from the worst point $\boldsymbol{V}_w$ to the centroid $\boldsymbol{C}$ is calculated as

$$\boldsymbol{W} = \boldsymbol{C} - \boldsymbol{V}_w \ . \tag{8.44}$$

A reflection of the worst point $\boldsymbol{V}_w$ is calculated, scaled by a reflection coefficient α

(by default, $\alpha = 0.75$)

$$\boldsymbol{R} = \boldsymbol{C} + \alpha \boldsymbol{W} \ , \tag{8.45}$$

and the fit function is evaluated at the reflection point $\boldsymbol{R}$. If the function value at the reflection point $\boldsymbol{R}$ is better than the best value of the remaining points, an expansion is considered, by calculating the reflection with an expansion coefficient γ (by default, $\gamma = 2$),

$$\boldsymbol{E} = \boldsymbol{C} + \gamma \boldsymbol{W} \ , \tag{8.46}$$

and the function is evaluated at that new point $\boldsymbol{E}$. If the expansion point $\boldsymbol{E}$ is better than the reflection point $\boldsymbol{R}$, the worst point in the simplex $\boldsymbol{V}_w$ is replaced by the expanded point $\boldsymbol{E}$, and the fitting continues. Otherwise, the reflection point $\boldsymbol{R}$ is used instead of $\boldsymbol{V}_w$, and the fitting continues. If the reflection value at $\boldsymbol{R}$ is better than the next worst value, the reflection point $\boldsymbol{R}$ replaces the worst point $\boldsymbol{V}_w$, and the fitting continues. If the reflection value at $\boldsymbol{R}$ is better than the worst value at $\boldsymbol{V}_w$, the reflection point $\boldsymbol{R}$ replaces the worst point $\boldsymbol{V}_w$ and the centroid position $\boldsymbol{C}$ is recalculated before further considerations. A contraction point $\boldsymbol{T}$ is then calculated by moving the worst point $\boldsymbol{V}_w$ by contraction coefficient β closer to the centroid $\boldsymbol{C}$ (by default, $\beta = 0.5$),

$$\boldsymbol{T} = \boldsymbol{C} - \beta \boldsymbol{W} \ , \tag{8.47}$$

and the function is evaluated. If the contraction point $\boldsymbol{T}$ is better than the worst point $\boldsymbol{V}_w$, it replaces the worst point, and the fitting continues. Otherwise, the whole simplex is shrunk around the best point $\boldsymbol{V}_b$ by a factor of two,

$$\boldsymbol{V}'_i = \frac{\boldsymbol{V}_b + \boldsymbol{V}_i}{2} \ , \quad i \neq b \ , \tag{8.48}$$

and the fitting continues.

Simplex fitting is performed using function

```
int
optimize_simplex(int   nparam,
                 float simplex[npoints][nparam+1],
                 float func(float params[nparam]),
                 bool  check(float *simplex[npoints][nparam+1],
                             int   ibest,
                             int   iworst));
```

where `nparam` is the number of fit parameters N (in our case—3), `simplex` array contains the simplex with `npoints` $=$ `nparam` $+ 1$ of point coordinates filled, each up to index `nparam`, and two functions called at each fitting step. Function `func()`, evaluates the fit function given the fit parameters `params`, and returns the evaluated value. Function `check()` checks the simplex contained in the `simplex` array for fit stop condition; it is passed the index of the best simplex point `ibest`, and the index of the worst simplex point `iworst`, and returns either TRUE, if fitting should be stopped, or FALSE, otherwise. Function `optimize_simplex()` returns the index in the `simplex` array of the best fit point, and `simplex` array contains the last simplex used during fitting.

The initial simplex is prepared by function

```
void
make_initial_simplex(int   dim,
                              float base_point[dim],
                              float corners[dim],
                              float *simplex);
```

where dim is the number of dimensions (parameters) N, base_point is the first base simplex point $\boldsymbol{V}_1$, and corners contains the displacement information $\boldsymbol{D}$ of each additional simplex point, one coordinate at a time for each of dim points in addition to the base point. This function returns filled simplex array simplex, ready to be passed to function optimize_simplex().

Function

```
bool
check_simplex(float *simplex, int ibest, int iworst);
```

is provided as an easy way to check for fit stop condition. This function ends fitting, if the distance between the best and the worst point is less than 1 in units given by the stop condition metric $\boldsymbol{M}$. Before calling this function, the caller must call

```
void
check_simplex_init(float *given_metric[dim]);
```

in order to provide the metric $\boldsymbol{M}$ for the stop condition test.

The simplex algorithm can be modified by changing fields in structure simplex_opt:

- alpha is the reflection coefficient, $\alpha > 0$. Default value is 0.75.

- gamma is the expansion coefficient, $\gamma > 0$. Default value is 2.0.

- `beta` is the contraction coefficient, $0 < \beta < 1$. Default value is 0.5.
- `flat_center` is a boolean flag that controls where the centroid of the simplex is located. FALSE means the centroid is center of all simplex points. TRUE means centroid is the center of simplex points except the worst point, i.e., it lies in the hyper-plane of the remaining points. The coefficients α, β, and γ significantly depend on the choice of this flag. Default value is TRUE.
- `max_iterations` specifies maximum number of iterations, after which the fitter gives up. Default value is 999.

8.3.3 Final Decision

In the χ^2 distribution for some events, there is a narrow minimum in the direction of the first ring, i.e., when the two rings almost completely overlap. This phenomenon occurs more strongly for electron events than for π^0 events.

In order to improve the discrimination between electrons and pions, a second fit is performed, following an identical procedure. The only differences are that the points of the starting simplex are chosen at the vicinity of the first ring, and the stopping condition requires somewhat higher precision of the result. The first point of the simplex is $\boldsymbol{V}_1 = (0, 0, 0.2)$, the remaining simplex corners are chosen from the values $\boldsymbol{D} = (5, 5, 0.3)$, and the metric for the stop condition is $\boldsymbol{M} = (1, 1, 0.01)$.

This fit is called a “forward” fit, because it concentrates in the forward direction defined by the first ring. The first fit is called a “wide” fit because the initial simplex spans quite a large area of the parameter space.

Finally, a decision is made about which fit is the best result. A value of 100 (chosen by trying which value gives good result) is added to the χ^2 of the forward fit and the fit with a smaller χ^2 value is chosen.

Energy fraction of the weaker ring is calculated:

$$f = \frac{E_2}{E_1 + E_2} \, , \tag{8.49}$$

where E_1 is the energy of the stronger ring, and E_2 is the energy of the weaker ring. In later analysis, only events with $f > 0.1$ are accepted as π^0 candidates.

In the fitting process, no minimum opening angle constraint is applied. Also, at this stage, there is no invariant mass constraint which is applied at later stages of the analysis.

Function

```
float
do_fit_pi0(float energy,
           float vertex[3],
           float dir[3],
           float eigvec[3][3],
           float fit_dir[2][3],
           float fit_e[2]);
```

performs π^0 fitting. This is the actual main function called to perform the fits. It takes total `energy` of the event, event `vertex`, direction of the first ring `dir`, and charge inertia tensor eigenvectors `eigenvec`. It sets the directions of the two gammas in fit_dir array, their energies in fit_e array, and returns the likelihood of the hypothesis of these two gammas. By default, the "wide" fit is performed. Before fitting, the fit kind can be selected by setting par_index field in fit_pi0_opt structure to 1 instead of the default value of 0. Structure fit_pi0_opt contains the following members:

- par_index chooses the index in the init_par array below. Default value is 0.
- init_par[NUM_INIT_PAR] contains the information about different fit kinds. NUM_INIT_PAR is 2. Each array element is the structure that contains the following elements, each of which is an array of fit parameters (ϕ, θ, f):
  - metric[NUM_FIT_PAR] array contains the stop condition metric $\boldsymbol{M}$.
  - start[NUM_FIT_PAR] array contains coordinates of the first simplex point $\boldsymbol{V}_1$.
  - corners[NUM_FIT_PAR] array contains the displacements of each remaining simplex corner with respect to the first simplex point $\boldsymbol{D}$.

  The first element in the array contains the "wide" fit parameters: metric ($\boldsymbol{M}$) = (2, 2, 0.02), start ($\boldsymbol{V}_1$) = (-30, 0, 0.4), and corners ($\boldsymbol{D}$) = (20, 20, 0.2). The second element in the array contains the "forward" fit parameters: metric ($\boldsymbol{M}$) = (1, 1, 0.01), start ($\boldsymbol{V}_1$) = (0, 0, 0.2), and corners ($\boldsymbol{D}$) = (5, 5, 0.3).

Function do_fit_pi0(), after performing timing cut by calling timing_cut(), invokes simplex optimization optimize_simplex() using lik_fit_pi0() as the evaluation function, and check_simplex() as the function that makes the decision about end of fitting, and finally converts the fit parameters to tracks by calling param_to_tracks().

Function

```
float
lik_fit_pi0(float param[NUM_FIT_PAR]);
```

is the fit evaluation function. First, it converts fit parameters to tracks by calling

```
void
param_to_tracks(float param[NUM_FIT_PAR],
                float energies[2],
                float directions[2][3]);
```

where param array contains the fit parameters. This function returns photon energies in array energies, and photon directions in array directions. After getting the track parameters, function lik_fit_pi0() calls track_likelihood() described in Section 8.2.

A reverse transformation function is also provided:

```
void
tracks_to_param(float reference_dir[3],
                float eigvec[3][3],
                float energies[2],
                float directions[2][3],
                int   closest_index,
                float param[NUM_FIT_PAR]);
```

where reference_dir is the direction of $\phi = 0$, $\theta = 0$ in fit parameters, eigenvec array contains the inertia tensor eigenvectors, energies array contains the energies of the two photons, `directions` array contains the directions of the two photons, and `closest_index` contains the index that specifies which photon is the main track, fit parameters are computed for the other one. This function returns the fit parameters in the `param` array.

8.4 Performance

Performance of the fitter was tested on Monte Carlo samples. Monoenergetic π^0 samples were generated uniformly in the Super-Kamiokande detector volume. The four samples have momenta of 200, 400, 600, and 800 MeV. Electron samples with energies corresponding to the π^0 visible energies were generates as well. The polfit fitter was applied to those samples.

Distributions of the invariant mass are shown on Figure 8.6. The π^0 peak is clearly visible. Invariant mass distributions of the electron sample are shown on Figure 8.7. As expected, most of the events reconstruct with a very low invariant mass. They can be easily removed with an invariant mass cut around the π^0 mass.

The samples were also fit with a standard fitter. Table 8.1 in the column labelled "atmpd" shows the percentage of events that were reconstructed as having two e-like rings and that fall within the π^0 invariant mass range. For events which were reconstructed as single e-like rings, the polfit fit results were examined. Those of them which passed the fitter successfully and reconstructed with invariant mass within the π^0 range, were added to "atmpd" events. The percentage of events reconstructed

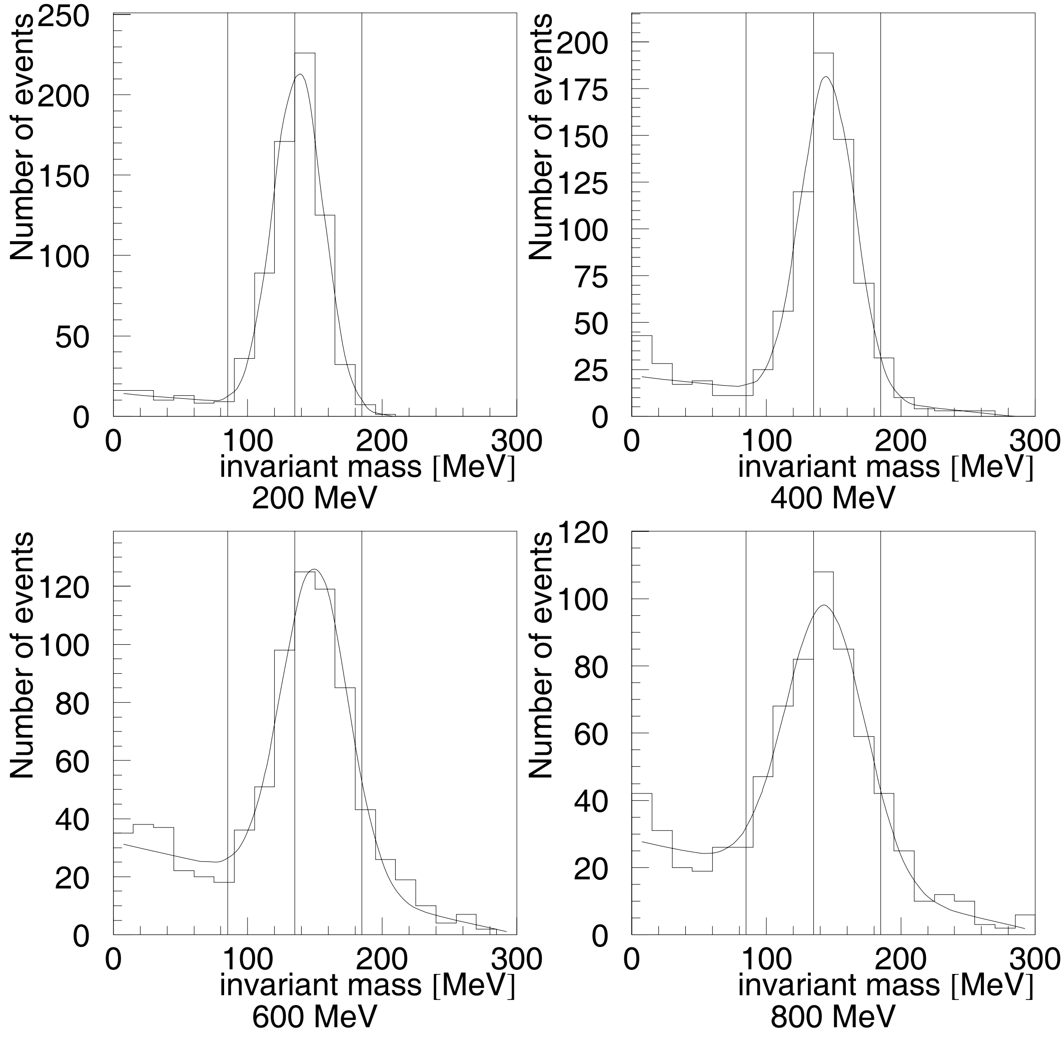


Figure 8.6: Invariant mass distributions for the pure π^0 samples with momenta of 200, 400, 600, and 800 MeV. The central vertical line marks the π^0 mass, the other two vertical lines mark an invariant mass range within 50 MeV of the π^0 mass. The curve shows a fit of a Gaussian with a linear background.

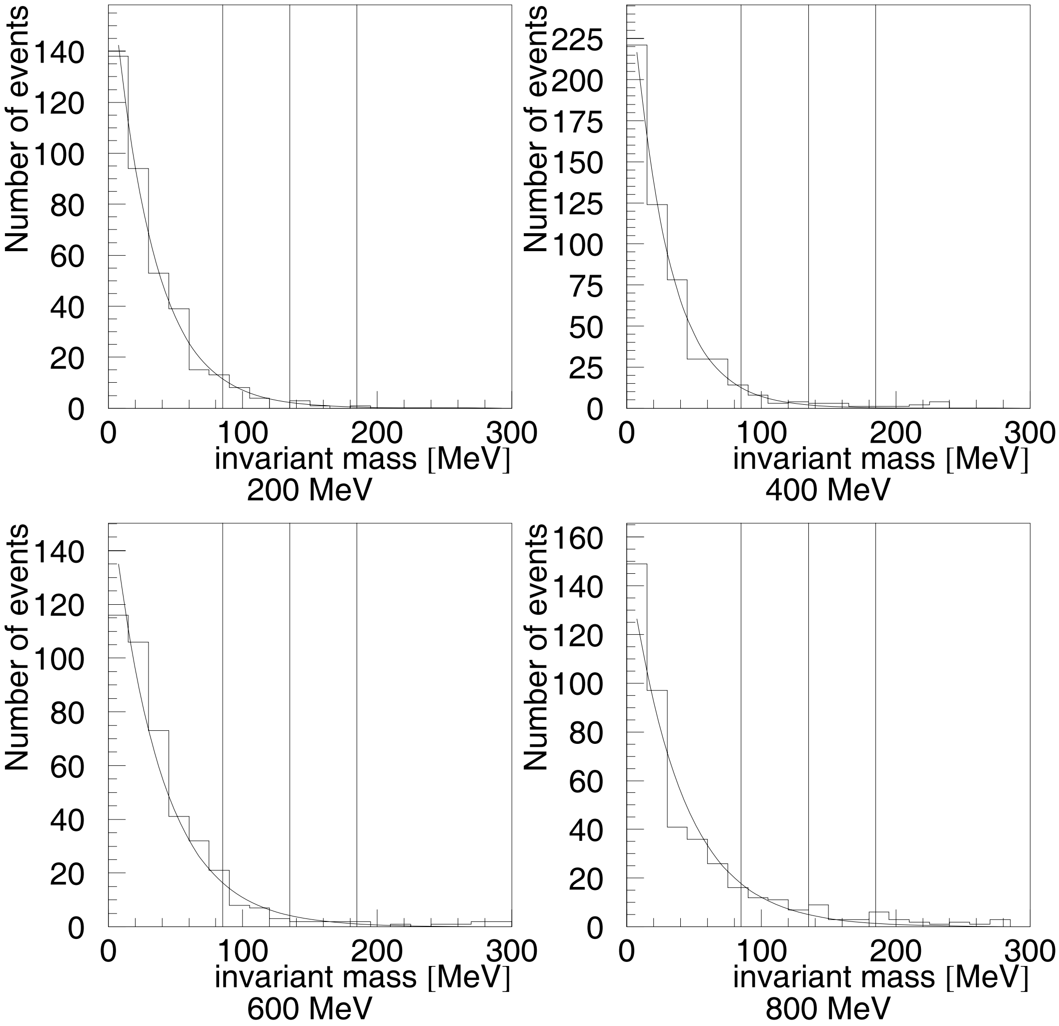


Figure 8.7: Invariant mass distributions for pure electron samples with momenta of 200, 400, 600, and 800 MeV. The central vertical line marks the π^0 mass, the other two vertical lines an mark invariant mass range within 50 MeV of the π^0 mass. The curve shows an exponential fit.

		atmpd	atmpd+polfit
π^0	200 MeV	66%	87%
π^0	400 MeV	59%	74%
π^0	600 MeV	41%	62%
π^0	800 MeV	19%	53%
e^-	200 MeV	2.1%	2.9%
e^-	400 MeV	1.7%	4.1%
e^-	600 MeV	0.9%	4.2%
e^-	800 MeV	0.7%	6.2%

Table 8.1: Percentage of events reconstructed in π^0 invariant mass range.

this way as π^0 is shown in the "atmpd+polfit" column. Reconstruction efficiency was increased significantly (see the π^0 rows), with some degradation of purity (see the e^- rows).

8.5 Interface

The code is designed in a modular way with logical parts that can be treated like black boxes with a well defined input and output and no side effects. Most of the elements in the libraries are easily reusable, i.e., they can be (and have been) used (by calling) by other people in their code. These parts include: raw light flux from various particle types, expected light generation for an arbitrary ring configuration, generation and use of custom patches, scattered light generation given an arbitrary direct light flux, charge normalization, likelihood (or χ^2) calculation, charge inertia tensor calculation, general simplex fitting, and, of course, the π^0 fitter itself.

Appendix A describes in detail the interface between C and Fortran that is utilized by this code.

Chapter 9

Data Samples

9.1 Atmospheric Neutrino Sample Selection

The fully contained atmospheric neutrino sample was collected for 1489.2 live detector days. The Monte Carlo contains 100 years of simulated data.

The fully contained sample is selected with the following cuts:

- Visible energy is greater than 30 MeV. This removes the low energy background.
- Largest OD cluster has fewer than 10 hits. This removes events which are not fully contained.
- Distance from the reconstructed vertex to the nearest wall is greater than 200 cm. This selects events within the fiducial volume.

In the single ring sample, lower momentum cuts are made at $p > 100$ MeV for electrons and $p > 200$ MeV for muons. These cuts depend on the reconstructed particle ID. The sample is further split into sub-GeV and and multi-GeV samples at a visible energy of 1.33 GeV. This boundary is used for historical reasons, in order to make it easier to compare Super-Kamiokande results with the Kamiokande experiment.

9.2 Atmospheric Neutrino Sample Characterization

Figures 9.1–9.5 show several distributions which characterize the fully contained sub-GeV sample. On all figures, the Monte Carlo (solid histogram) is normalized to the livetime and has no neutrino oscillations applied. The observed differences between measured data and Monte Carlo are caused by an uncertainty of the total neutrino flux and by a partial disappearance of muon neutrinos due to neutrino oscillations.

Fig. 9.1 shows the number of rings distribution. Single ring events are 67% of the sample, two-ring events are 19%. Figures 9.2 and 9.3 show no evidence for contamination from entering events within the fiducial volume. Fig. 9.4 shows the momentum spectra. For the multiple ring events momenta vectors (based on PID) of all particles were added to form a total vector—its magnitude is plotted. Figure 9.5 shows the distribution of the particle ID likelihood $\sqrt{P_\mu} - \sqrt{P_e}$. We can see a very good particle

ID separation.

9.3 Multiple Ring Sample

The main neutrino oscillation analysis at Super-Kamiokande concentrates on single ring events. It is designed to select charged current quasielastic events $\nu_l N \to l N'$.

Multiple ring events are more difficult to analyze compared to single ring events. This happens due to variety of reasons.

One reason is the size of the samples used for analysis. Single rings comprise 67% of the data set (Fig. 9.1). Practically all of the events can be used for analysis. Multiple ring events are not only less frequent, but typically we also have to apply more cuts which restrict the size of the sample even further.

Another reason is that multiple ring events are more difficult to reconstruct than single ring events. One has to recognize the additional rings. They are, frequently, faint compared to the main ring and overwhelmed by direct and scattered light from the primary track. The amount of light assigned to each ring has to be correctly divided between all the rings, taking into account scattered light properly. The particle type of each ring has to be identified, which is complicated by scattered light from other rings and overlap with other rings.

Additionally, cross-sections of the reactions which contribute significantly to the multiple ring events are less precisely known than quasi-elastic cross-sections.

Contributions of reaction channels to the samples with various number of rings are

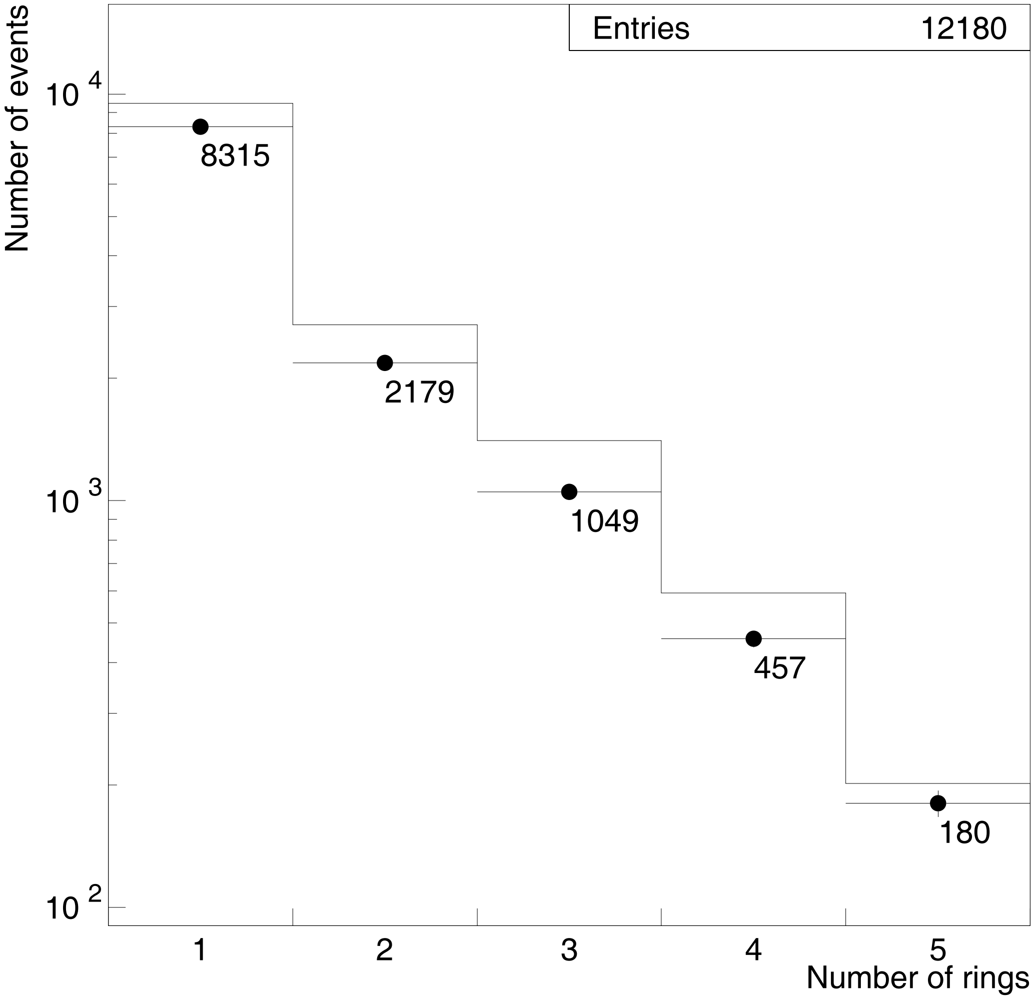


Figure 9.1: The number of reconstructed rings in the fully contained sample. Black circles show the data, histogram shows the the unoscillated Monte Carlo.

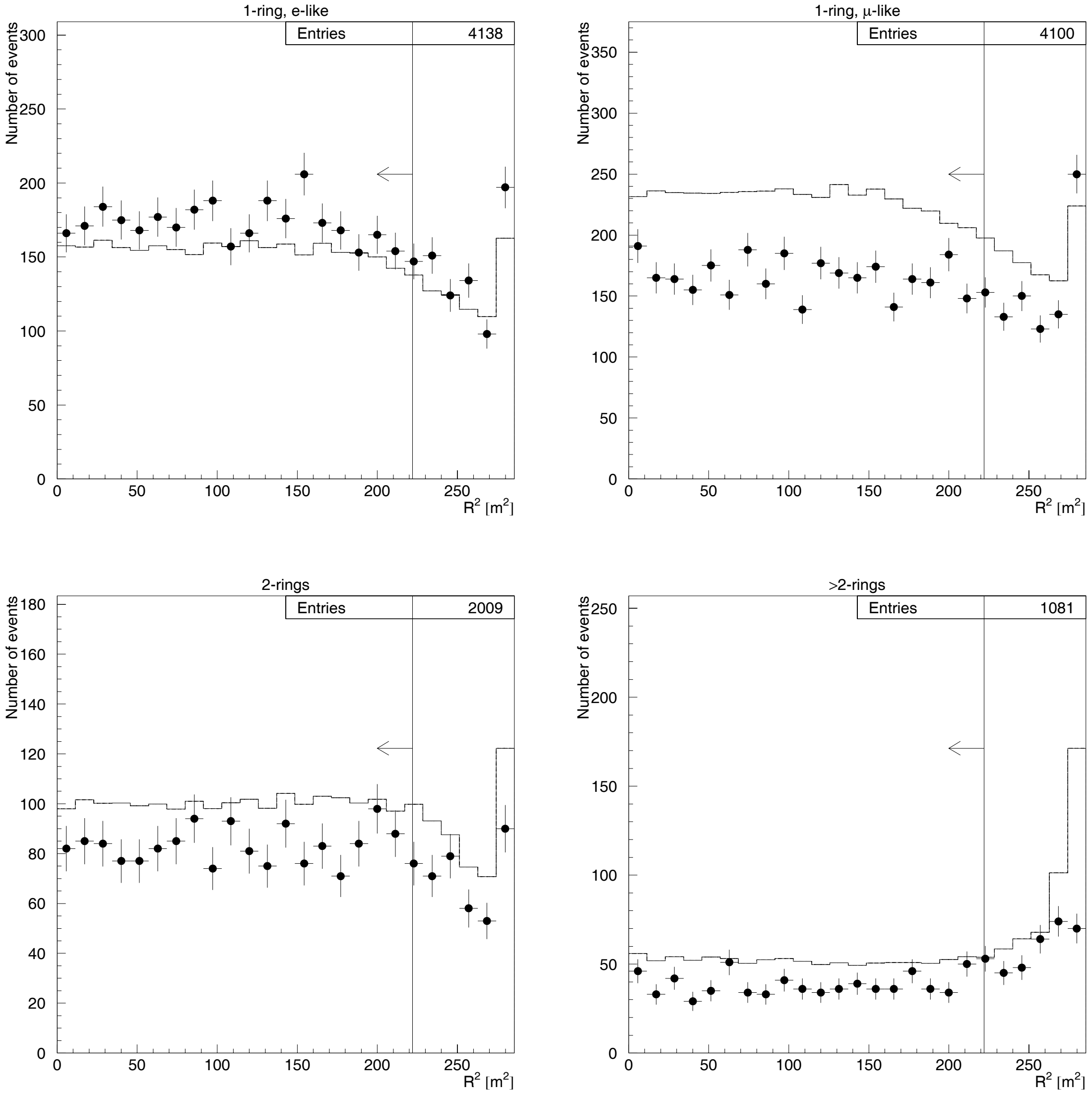


Figure 9.2: R^2 distributions of the Sub-GeV sample. 0 is at the center of the tank. Z is limited to the fiducial volume. Standard lower momentum cuts were applied to 1-ring e-like and μ-like samples. No lower momentum cuts were applied to multiple ring samples. Lines show the fiducial volume cut. Points with error bars are measured data. Solid histogram is the Monte Carlo normalized by livetime.

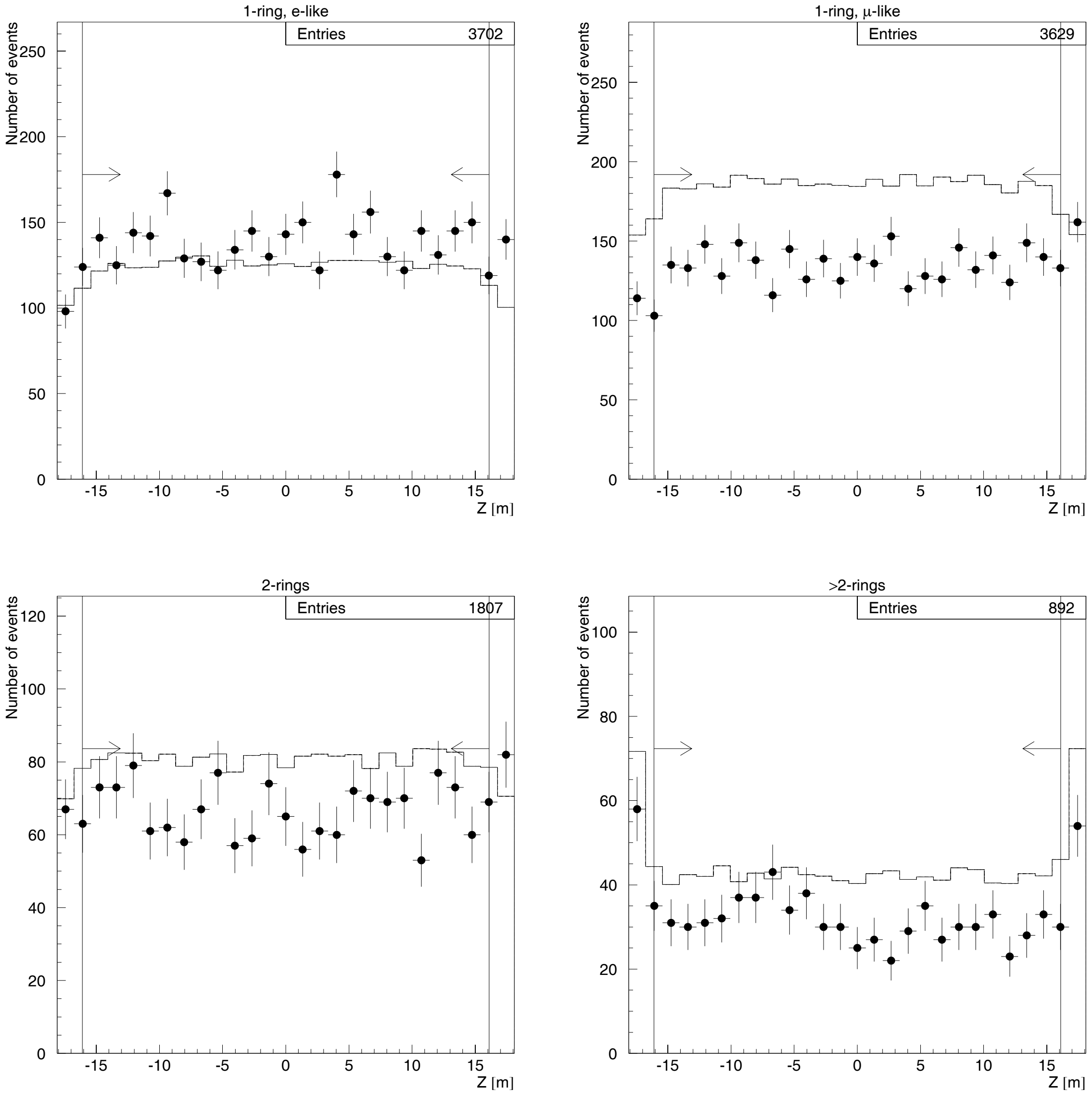


Figure 9.3: Z distributions of the Sub-GeV sample. Bottom of the tank is at the negative end of the histogram. R is limited to fiducial volume. Standard lower momentum cuts were applied to 1-ring e-like and μ-like samples. No lower momentum cuts were applied to multiple ring samples. Lines show fiducial volume cut. Points with error bars are measured data. Solid histogram is the Monte Carlo normalized by livetime.

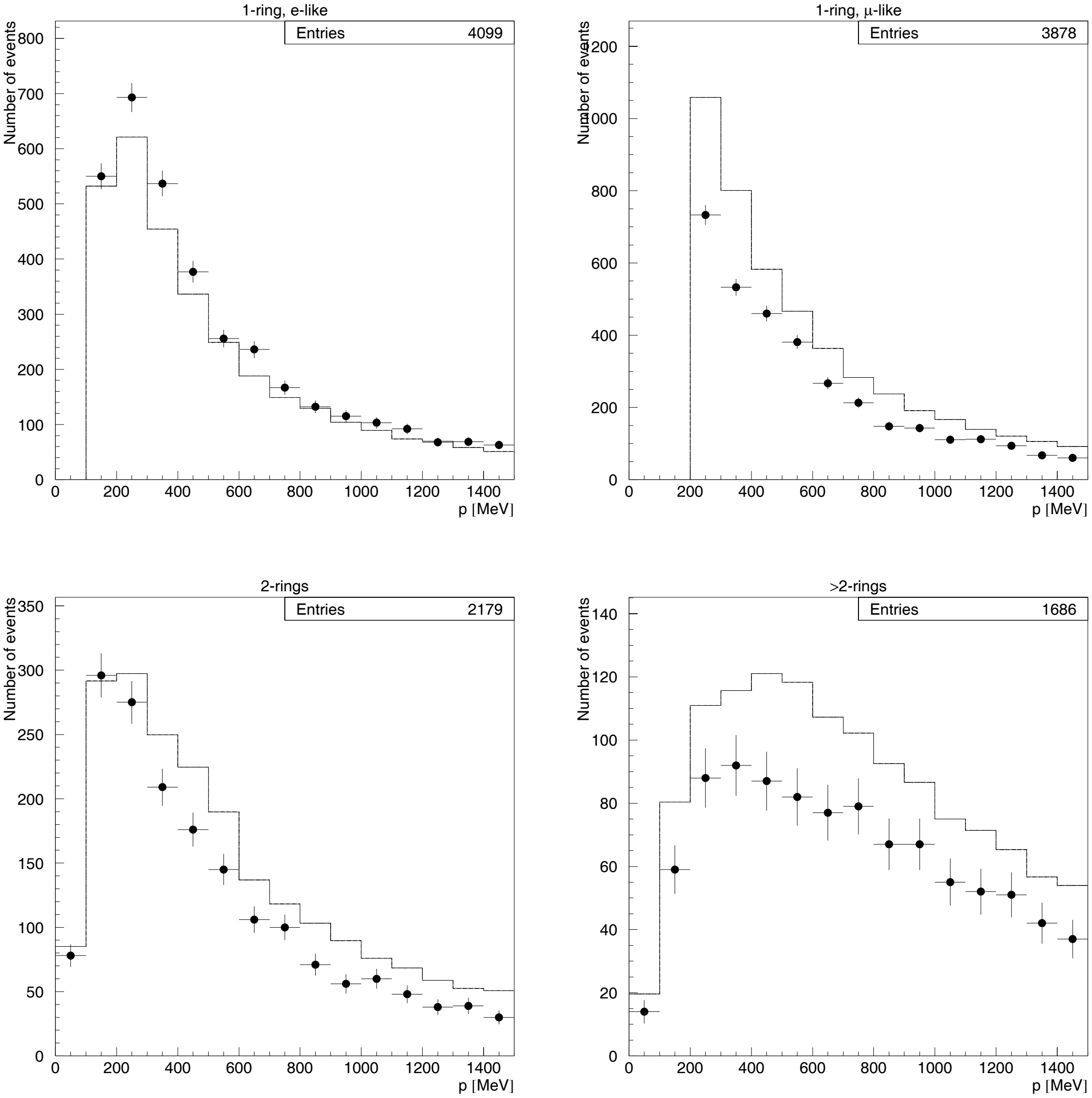


Figure 9.4: Total momentum distributions. No upper visible energy cut was applied. Standard lower momentum cuts were applied to 1-ring e-like and μ-like samples. No lower momentum cuts were applied to multiple ring samples. Points with error bars are measured data. Solid histogram is the Monte Carlo normalized by livetime.

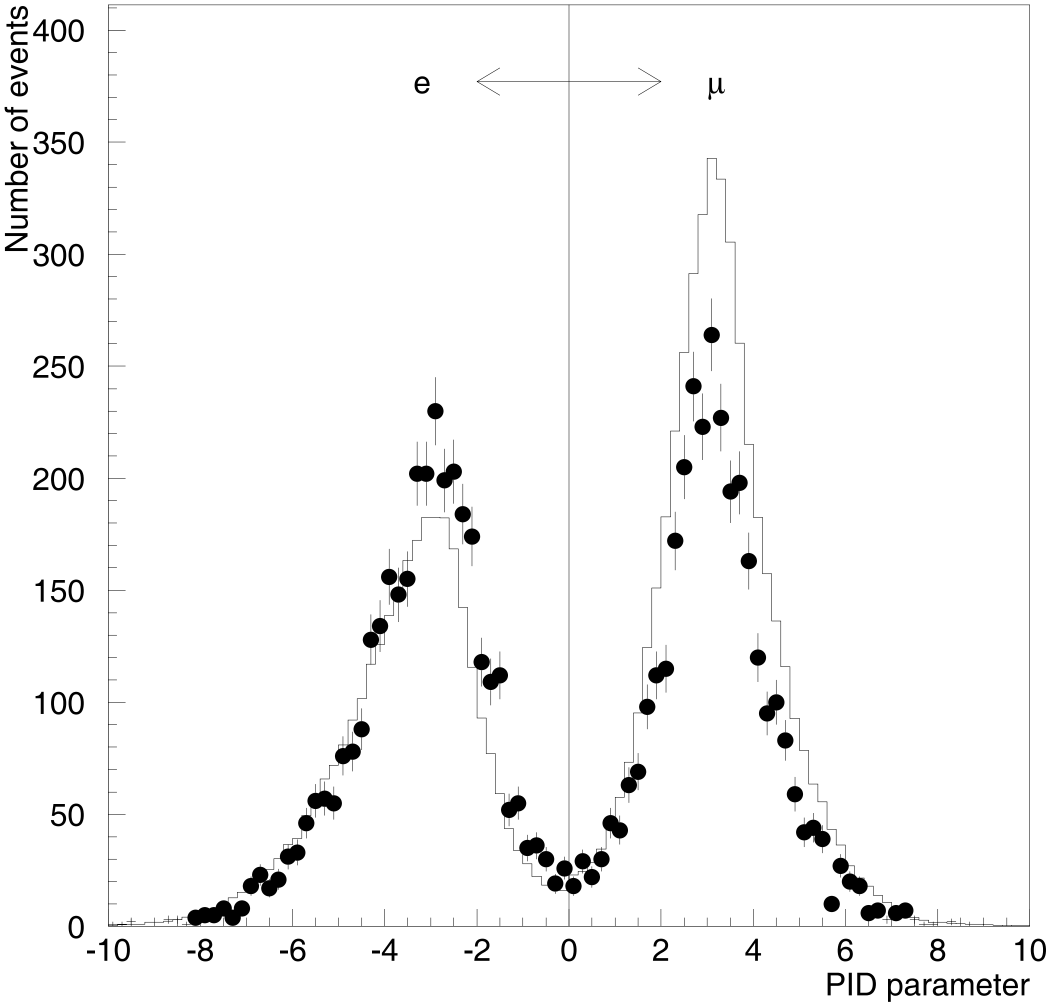


Figure 9.5: Particle ID parameter distribution for the 1-ring sub-GeV sample. Points with error bars are measured data. Solid histogram is Monte Carlo normalized by livetime. Tracks with positive PID are classified as μ-like. Tracks with negative PID are classified as e-like.

shown on Figure 9.6. Single ring events are dominated by quasielastic events, followed by charged current single pion production. Two ring events consist mostly of various pion production events with the largest contribution from single pion production. Three ring events have an increased multi-pion production rate compared to single-pion channels. Also, deep inelastic scattering events contribute significantly. Events with more than three rings are dominated by deep inelastic scattering followed by multiple pion production.

The above breakdown shows that frequently the number of rings does not correspond to the number of observable particles produced in the reactions (e.g., one for each $l^{\pm}$ or $\pi^{\pm}$, two for each π^{0}). Slow pions and muons can be under Cherenkov threshold. Recoil protons are usually under the Cherenkov threshold. These particles are not observed. Pions can interact in the nucleus in which they were produced, for example, undergoing absorption or charge exchange. Even when they manage to escape the nucleus, they still can interact with the surrounding water (nuclei of oxygen and protons). Occasionally, a particle can scatter strongly and produce two rings. Even when a particle produces a Cherenkov ring which, when alone, would be observed, it can be missed if overlaps strongly with another ring, or it can be overwhelmed by direct and indirect light from brighter tracks.

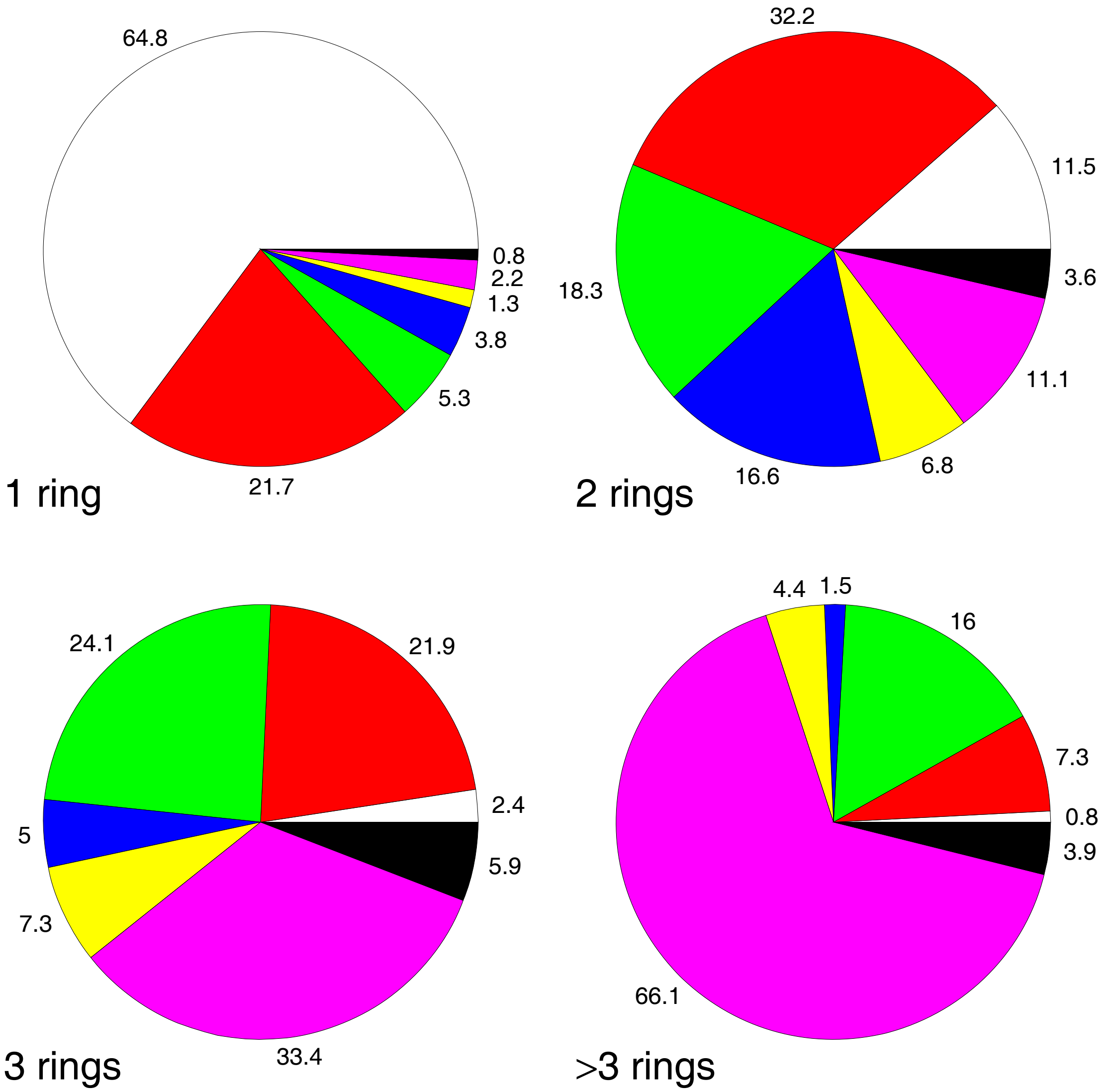


Figure 9.6: Composition of samples with various numbers of rings. Monte Carlo interaction categories are shown starting at 3 o'clock position and going counter-clockwise: CC quasielastic (white), CC single π production, CC multi π production, NC single π production, NC multi π production, deep inelastic scattering, and other (black). Single pion production includes: pion from delta resonance, coherent production, and diffractive production. Other interactions include: kaon and eta production (both CC and NC), and NC elastic interactions. Numbers show percentages of each interaction mode. Data set used is fully contained unoscillated MC sample (no upper energy cut was explicitly applied).

9.4 Two Ring Sample

Of multiple ring events, two ring events are easier to analyze than events with higher multiplicities, and they are most abundant.

π^0 events are two ring events. There is a relatively large uncertainty in predicting cross section for these events. However, Vissani and Smirnov [39] proposed that by comparing the rate of π^0 events to those of other two ring events the systematic error can be reduced, because cross-sections for those events are related and systematic errors partially cancel when taking the ratios.

Another reason for looking at two ring events is to observe the effects of neutrino oscillations in the multiple ring sample. For this purpose a series of cuts was developed which defines two samples: one, mostly due to muon neutrino interactions, and another one, mostly due to electron neutrino interactions. The cuts were designed to yield samples of high purity with reasonable statistics. Another sample defined, predominantly due to neutral current interactions, is the π^0 sample.

9.5 Two Ring Sub-Samples

We start with the sample of reconstructed atmospheric neutrinos. The first set of cuts is analogous to the sub-GeV sample cuts. The requirement `nhitac` < 10 selects fully contained events. Events within the fiducial volume are selected by requiring the reconstructed vertex to be further than 2 m from any wall. The visible energy is required to be smaller than 1330 eV. This selects events in the sub-GeV range. There is no lower energy bound at this point, unlike in the standard analysis that cuts at 30 MeV of visible energy. Later, a momentum cut is performed instead. Of course, the number of reconstructed rings is required to be two.

Based on the reconstructed particle ID of each of the two rings, the sample is split into three parts. Each ring is called either `M`, if it was reconstructed as a muon-like (also called non-showering) ring, or it is called `S`, if it was reconstructed as an electron-like (also called showering) ring. The three samples are called `MM`, `SS`, and `MS`.

The `MS` sample is further split into two samples `Ms` and `Sm`, based on the reconstructed momentum of the two rings. An event is assigned to the `Ms` sample when the momentum of the non-showering track is larger than the momentum of the other track, otherwise it is assigned to the `Sm` sample. Note, that momentum reconstruction depends on the reconstructed particle ID.

Within the `SS` sample, there are many π^0 events. Events with an invariant mass within 50 MeV from the π^0 mass of 135 MeV are assigned to the π^0 sample. The remaining sample is called $\texttt{SS} - \pi^0$ (`SS` minus pi zero).

To improve the purity of the samples, a cut was made on the number of, reconstructed electrons which came from muon decays (i.e., the number of reconstructed muon decays). In the `MM` and the `Ms` samples, only the events with one or more muon decays were accepted. In the `SS` (including the π^0) and the `Sm` samples, only the

	ν_μ CC	ν_e CC	NC	data
MM	169.7	1.1	15.7	101
Ms	204.4	10.6	16.1	159
Sm	7.6	29.7	11.1	39
SS $-\pi^0$	13.8	129.8	69.5	183
π^0	7.2	37.8	155.6	181

Table 9.1: Composition of two ring samples. Number of Monte Carlo events is normalized to detector live time. Number of observed events is also shown in the last column. Muon decay electron cuts are performed on this sample, and only events with total momentum over 300 MeV are included.

events with no muon decays were accepted. Composition of the samples is shown in Table 9.1.

The samples MM and Ms are combined to form a ν_μ-enriched sample. The samples SS $-\pi^0$ and Sm form a ν_e-enriched sample. The π^0 sample is the neutral current (NC) sample.

The total momentum vector is defined by adding the momentum vectors of each of the tracks. Direction of the event is defined to be the direction of the total momentum.

To improve the correlation between the direction of the incoming neutrino and the direction of the event, a momentum cut is performed. Only events with a total momentum above 300 MeV are accepted (Fig. 9.7).

The samples described here are used for further analysis described in the subsequent chapters.

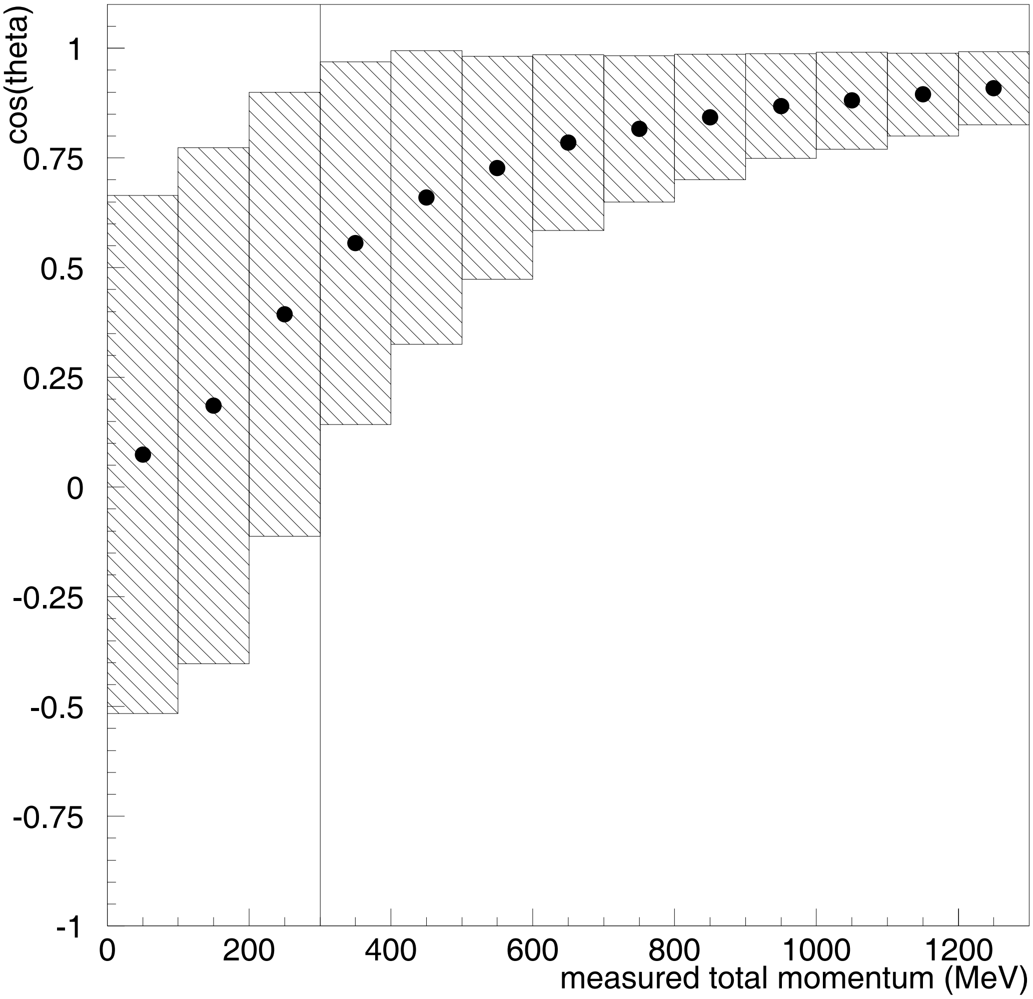


Figure 9.7: Correlation between the neutrino direction and the reconstructed direction. θ is the angle between the neutrino direction and the direction of the total reconstructed momentum. The sample is the set of fully contained sub-GeV two ring MC events within the fiducial volume. No further cuts were performed. The hatched boxes show the spread of the variables, the dots show the mean value. The vertical line shows the cut value. $\cos\theta = 0$ corresponds to no directional correlation, $\cos\theta = 1$ corresponds to perfect correlation.

Chapter 10

Analysis and Results

10.1 The Two Ring Samples

Two of the two ring samples (ν_e-enriched sample and ν_μ-enriched sample) were designed to select events produced by a specified neutrino flavor: an electron or a muon type. It is interesting to look at them to see if we can find the same distortion of the zenith angle distributions as seen in the single ring samples.

The distributions are shown in Fig. 10.1 and Fig. 10.2. The electron neutrino distribution is consistent with the unoscillated Monte Carlo. The muon neutrino distribution shows similar distortion as the single ring sample. The oscillated MC is shown as well. The oscillation parameters are chosen to be the same as the best fit value in the standard analysis. No fitting was performed here.

Oscillating the Monte Carlo is performed by re-weighting the events. First, all

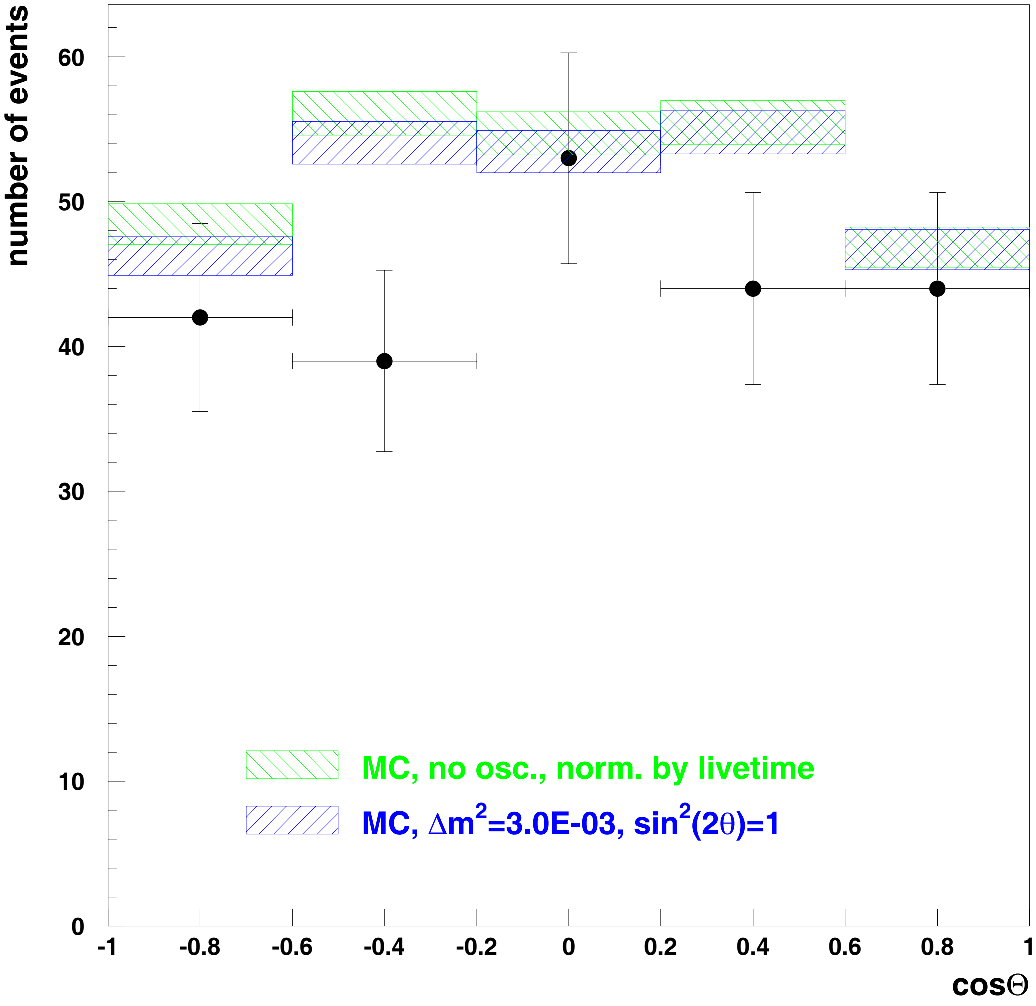


Figure 10.1: The two ring ν_e enriched sample. The points with the error bars show the measured data. The hatched boxes show the unoscillated and the oscillated Monte Carlo. The height of the boxes corresponds to the statistical error. The MC is normalized by the livetime. $\cos\theta = -1$ corresponds to up-going events, $\cos\theta = 1$ corresponds to down-going events.

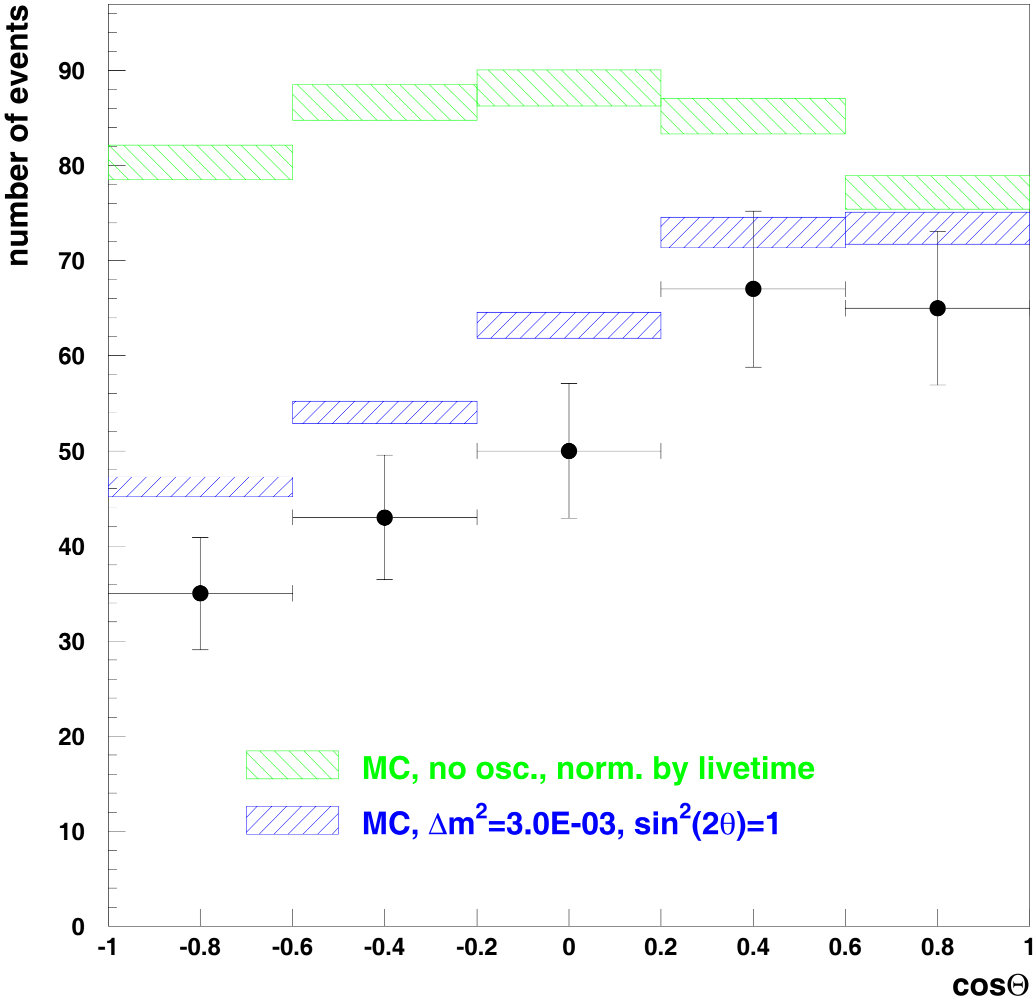


Figure 10.2: The two ring ν_μ enriched sample. The points with the error bars show the measured data. The hatched boxes show the unoscillated and the oscillated Monte Carlo. The height of the boxes corresponds to the statistical error. The MC is normalized by the livetime. $\cos\theta = -1$ corresponds to up-going events, $\cos\theta = 1$ corresponds to down-going events.

charged current interactions caused by ν_μ are identified. Each of them is assigned a weight between 0 and 1, which is the oscillation survival probability:

$$p = 1 - \sin^2 2\theta \, \sin^2(1.27 \, \Delta m^2 \, L/E) \; . \tag{10.1}$$

θ and Δm^2 are the mixing angle and the difference of the squared neutrino masses. They are the oscillation parameters. E is the energy of the incoming neutrino. L is the distance between the points of the neutrino production and the detection. L is calculated from the zenith angle of the incoming neutrino. The neutrino production height is assumed to be fixed at 18 km above the Earth's surface. The detector is assumed to be at the surface of a spherical Earth. When the argument of the second $\sin^2$ (the phase) is large, a small change of L or E can cause a large change in p. In this case the $\sin^2$ equals 0.5 on average. In the weight calculation, the $\sin^2$ is thus replaced with 0.5 when the phase is large.

We can ask the question—what Δm^2 best fits the observed shape of the ν_μ-enriched sample, or even better—what are the allowed values for Δm^2. This can be performed by minimizing χ^2 of the oscillated Monte Carlo on the ν_μ-enriched plot. The overall normalization is not very well determined. We can normalize the MC by the number of events, which is equivalent to checking only the shape of the distribution. Figure 10.3 shows the χ^2 distribution as a function of Δm^2 with the 68% and 90% confidence level allowed regions marked.

To illustrate the effect of chosen normalization, Figure 10.4 shows the same distributions but for the Monte Carlo normalized by livetime. It is apparent that the allowed Δm^2 range is sensitive to the choice of normalization.

While the overall normalization is the greatest source of uncertainty (and, therefore, we don't constrain it at all to the value predicted by the Monte Carlo), there are other (less significant) systematic errors present [29]. There are uncertainties in neutrino flux (composition, spectrum, production height), and particularly up-down asymmetry. It is very small because the flux is symmetric, except for the lowest energies due to geomagnetic effects, but in this region directional correlation of particles produced in SK is poor. Systematic uncertainties in neutrino interaction include relative cross-sections of the various interaction types, model parameters, and nuclear effects. Uncertainties in reduction and reconstruction include efficiency, background, ring separation, particle identification, energy calibration, and any up-down asymmetry of the detector (for example, of energy calibration). In order to fully treat the systematic errors, a term for each contribution needs to be added to the overall χ^2 and included in the minimization. The detailed determination of the systematic errors for this analysis is beyond the scope of this thesis.

10.2 π^0 Analysis

The polfit fitter was applied to the atmospheric neutrino sample which was reconstructed as single e-like ring events in order to recover some of the neutral current π^0 events, for which the standard analysis failed to find the second ring. Those π^0 events are expected to be difficult to fit, because the standard fitter failed. The results are

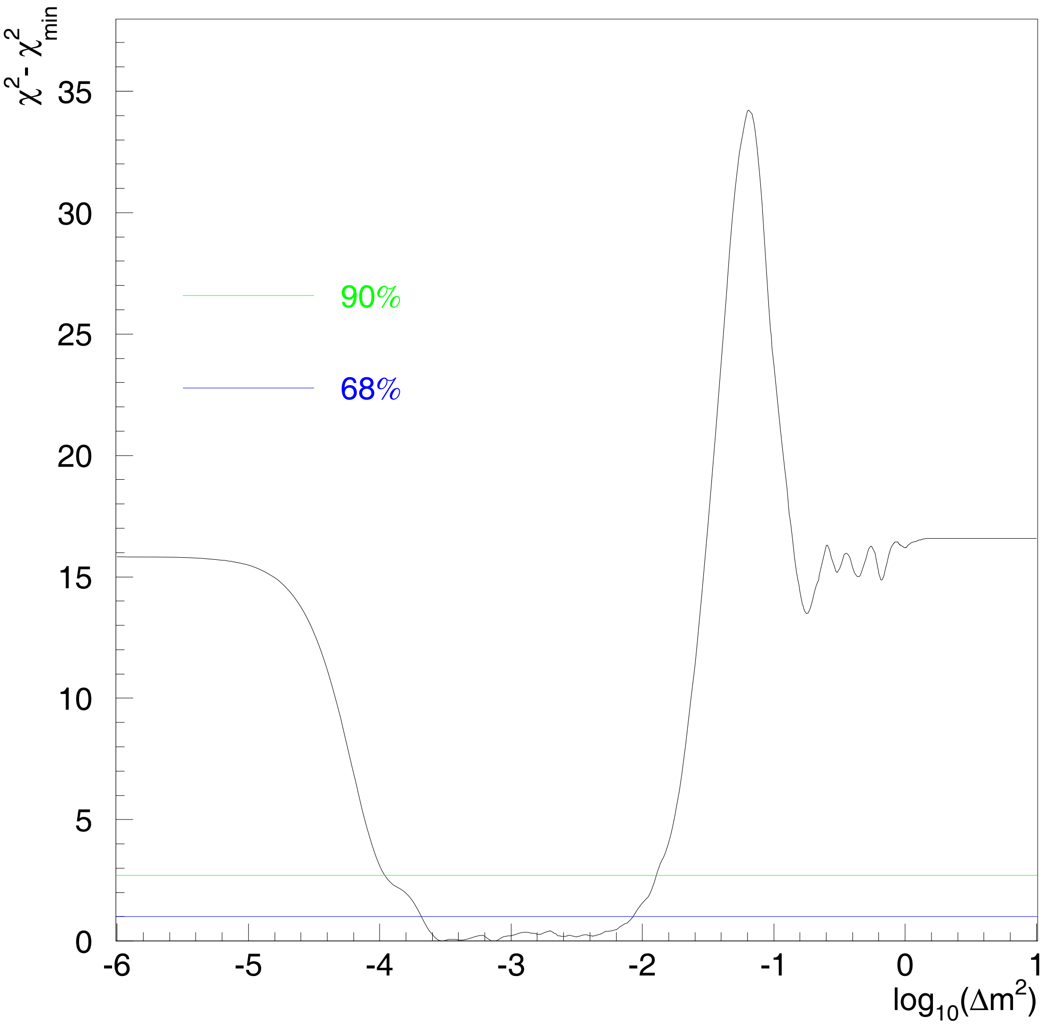


Figure 10.3: χ^2 distribution for the two ring muon neutrino enriched sample. The MC is normalized by the number of events. Δm^2 is measured in eV^2. The 68% and 98% confidence level regions are where χ^2 is below the corresponding horizontal lines.

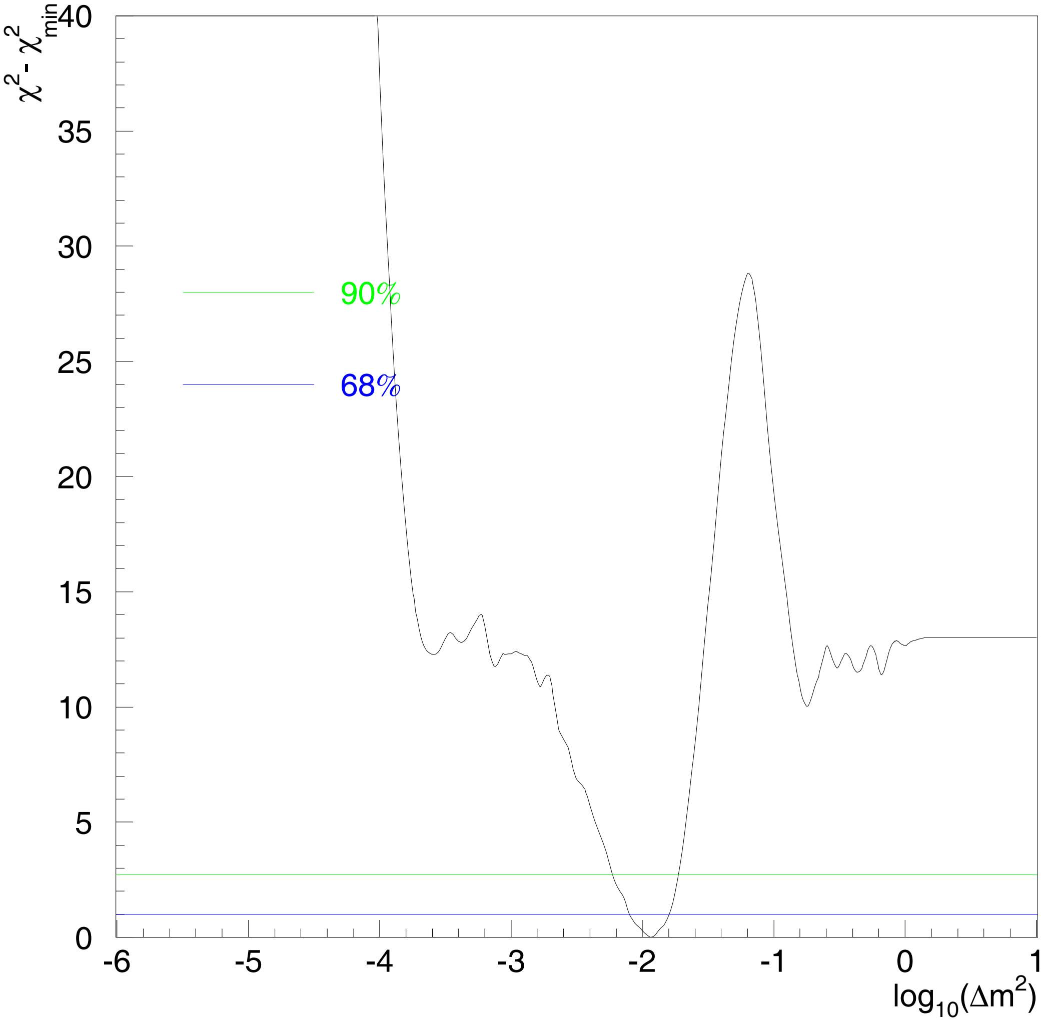
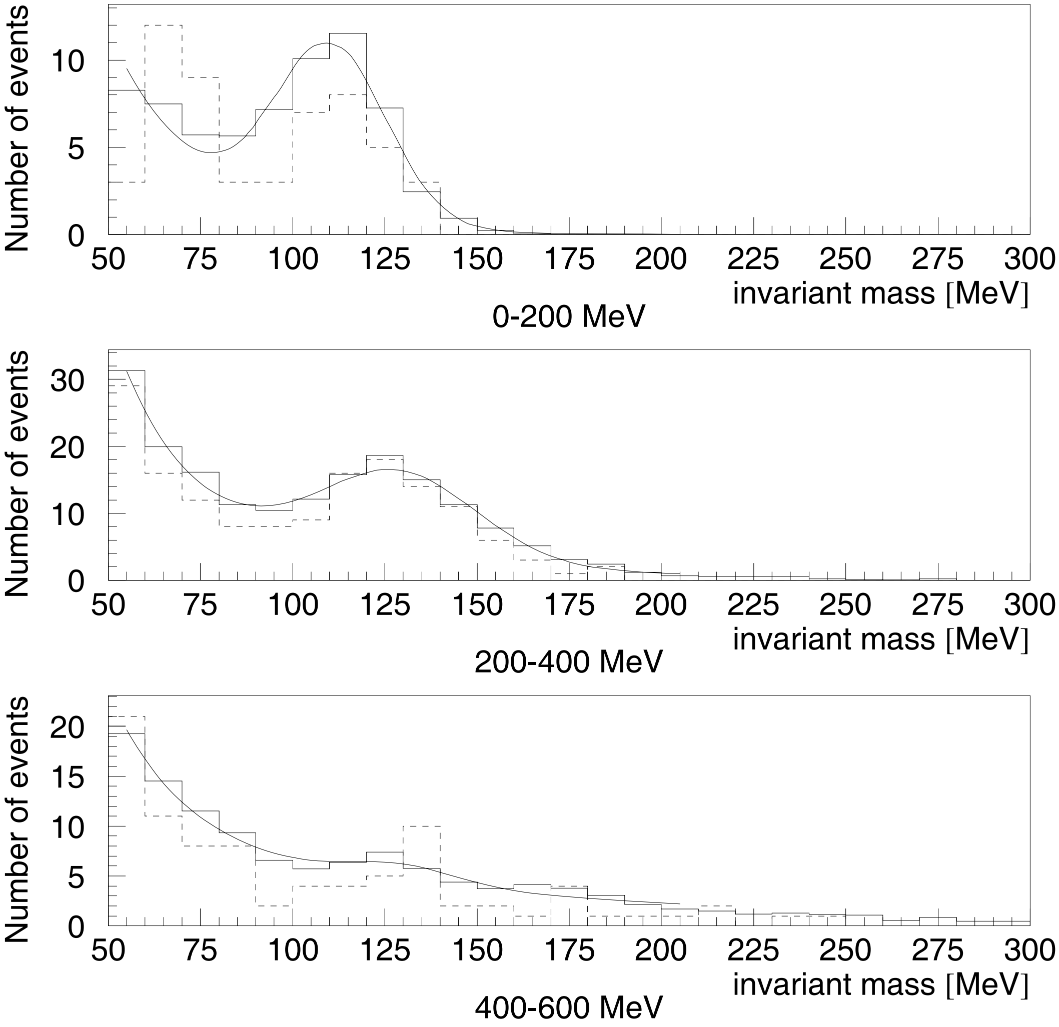


Figure 10.4: χ^2 distribution for the two ring muon neutrino enriched sample. The MC is normalized by the livetime (unlike the previous figure). Δm^2 is measured in eV^2. The 68% and 98% confidence level regions are where χ^2 is below the corresponding horizontal lines.

shown on Figure 10.5. The sample was subdivided into subsamples based on a visible energy: 0–200 MeV, 200–400 MeV, and 400–600 MeV. One still can see a π^0 peak in the invariant mass distribution. The background at the low invariant mass area is caused by the polfit fitter always accepting the second ring hypothesis, unless the energy fraction of the second ring $f < 0.1$.

Figure 10.6 shows the invariant mass distribution for combined two e-like ring sample consisting of two parts: the standard fitter, and the sample produced by the polfit fitter run on the single e-like ring sample. The sample consists of events in the sub-GeV visible energy range. For further analysis, all events within 50 MeV range around the π^0 mass of 135 MeV are considered to be π^0 events.

To study the efficiency of π^0 reconstruction, it is interesting to consider a distribution of $\cos\theta_{\mathrm{cms}}$, which is cosine of the angle between π^0 momentum and the direction of the two gammas in the center-of-mass reference frame. $\cos\theta_{\mathrm{cms}} \approx 0$ corresponds to the two gammas being produced perpendicularly to the π^0 momentum. In the detector reference frame, this is the case when the two observed gammas have equal energy and they are closest to each other (the largest overlap the the rings). $\cos\theta_{\mathrm{cms}} \approx 1$ corresponds to the two gammas being produced parallel to the π^0 momentum. In such case in the detector reference frame one of the gamma rings is the weakest and it is the furthest from the stronger ring. The distribution should be flat. Figure 10.7 shows the distributions for π^0 events with reconstructed momentum below 300 MeV, and Figure 10.8 shows the distributions for $p > 300$ MeV. Addition of events fit



0-200 MeV

200-400 MeV

400-600 MeV

Figure 10.5: Invariant mass distributions for the atmospheric neutrino sample. Three visible energy ranges are shown. The sample consists of events fit by polfit, which were fit as single e-like rings by the standard fitter. The solid histogram shows the Monte Carlo normalized by the livetime, the dashed histogram shows the measured data. The curve shows the fit of a Gaussian with two exponentials.

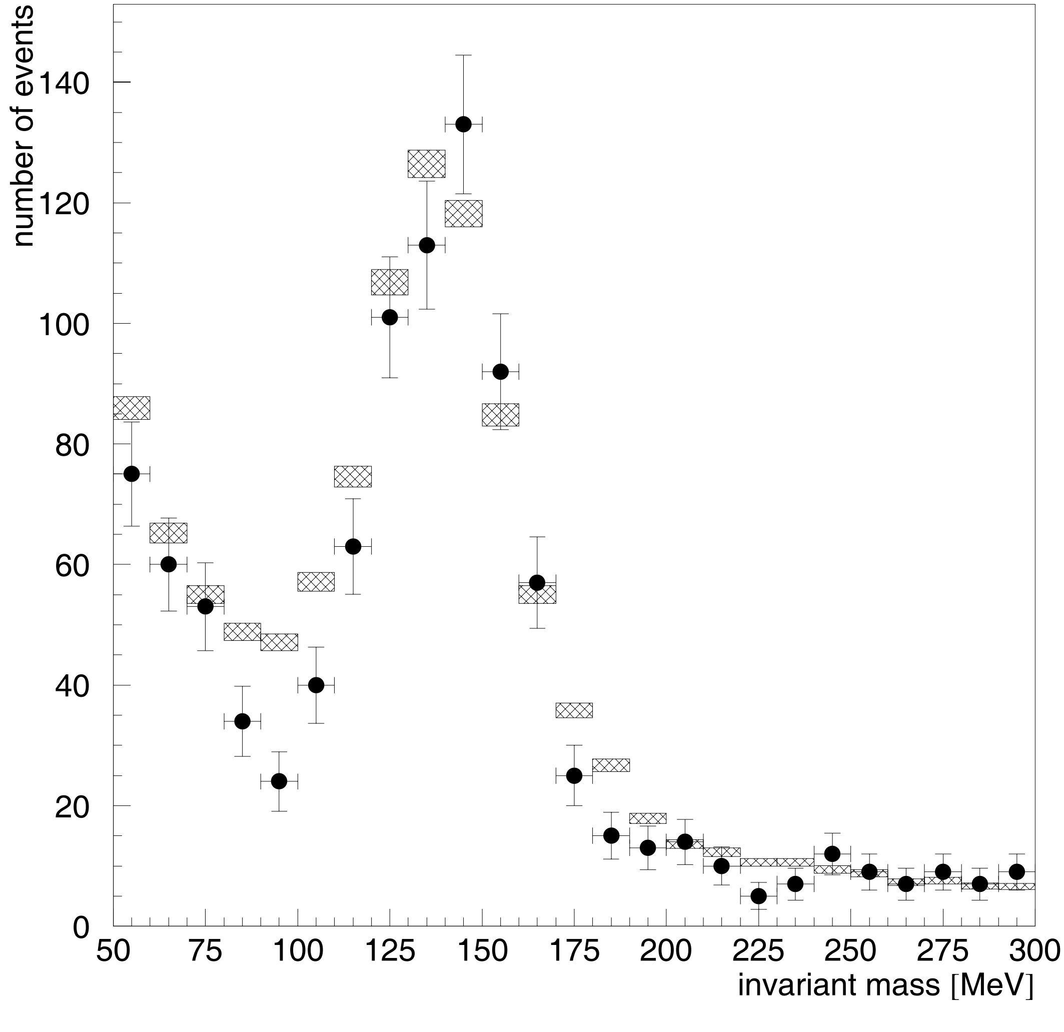


Figure 10.6: Invariant mass distribution for combined two ring *e*-like sample that includes polfit sample. Points show data, boxes show MC.

by polfit, which the standard fitter classified as single showering rings, flattens the distribution. Polfit picks up events missed by the standard fitter. For higher energy events the cut

$$f = \frac{E_2}{E_1 + E_2} > 0.1 \tag{10.2}$$

reduces number of events in the right-most bins.

The momentum distribution of the π^0 events is shown on Figure 10.9. The spectrum falls quickly for high momentum values. This is unfortunate, because the high momentum events are the most valuable. The low momentum events don't have a good directional correlation with the neutrino direction, see Fig. 9.7.

Figure 10.10 shows the zenith angle distribution of the π^0 events. Also shown, are three Monte Carlo distributions corresponding to: no oscillations, $\nu_\mu - \nu_\tau$ oscillations, and $\nu_\mu - \nu_s$ oscillations. The oscillation parameters used are $\Delta m^2 = 3 \times 10^{-3}$, and $\sin^2 2\theta = 1$. It is visible how $\nu_\mu - \nu_s$ oscillations decrease the overall π^0 event rate. Also, the shape of the zenith angle distribution changes in such case. The up-going events are more suppressed than the down-going events.

To emphasize the shape distortion a momentum cut has been applied. Only the events with total $p > 300$ MeV are accepted. The result is shown on Figure 10.11. Clearly, the rejection of the events with a low momentum, which do not carry much neutrino direction information, has helped emphasize the shape change. However, of course, the statistics suffers.

To help improve the statistical power, the events can be categorized in two bins

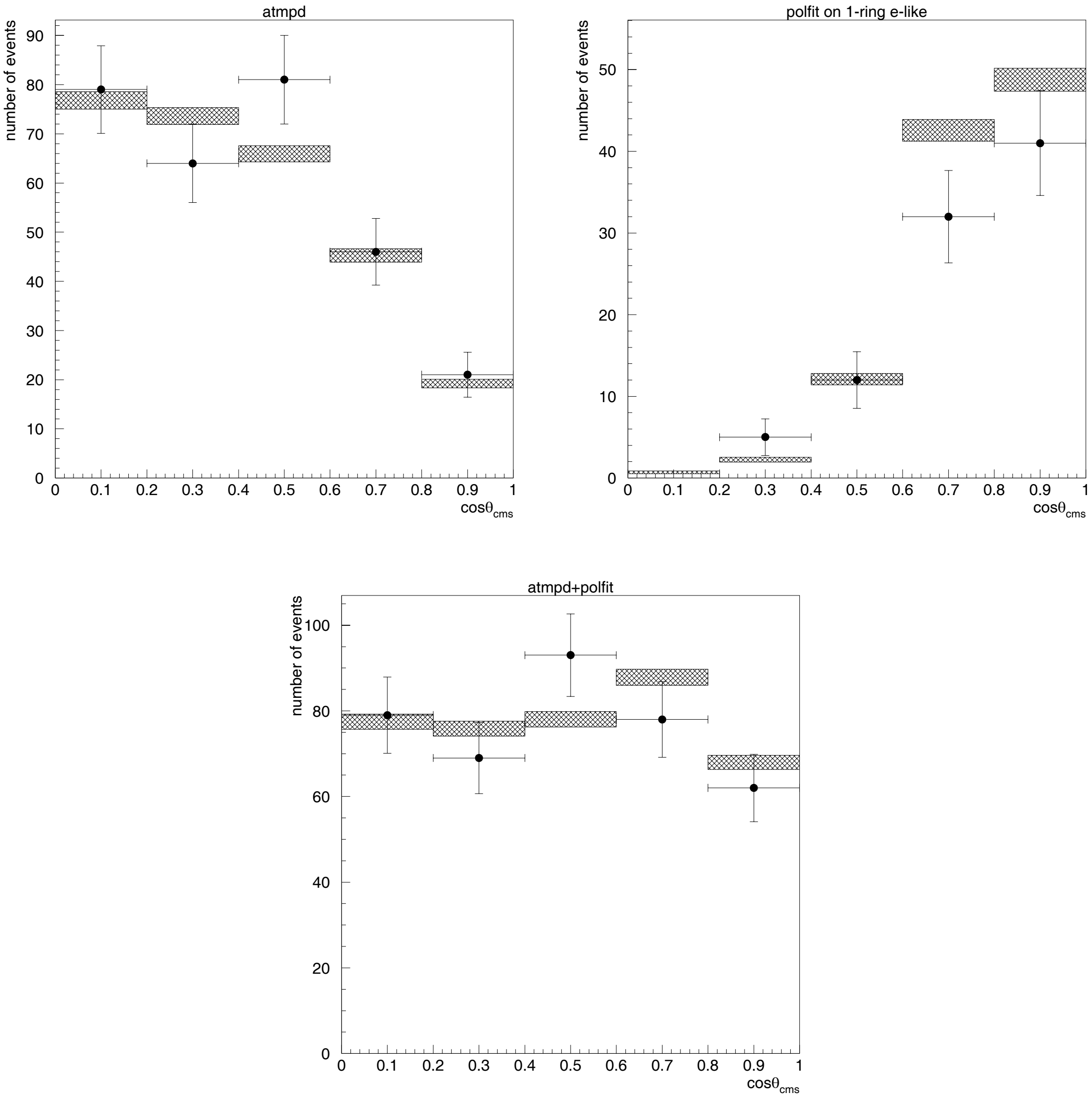


Figure 10.7: $\cos\theta_{\rm cms}$ distributions for π^0 with $p < 300$ MeV. Left plot shows events reconstructed by the standard fitter. Right plot shows the events reconstructed by polfit, which the standard fitter reconstructed as single showering rings. Bottom plot is a sum of the two top plots.

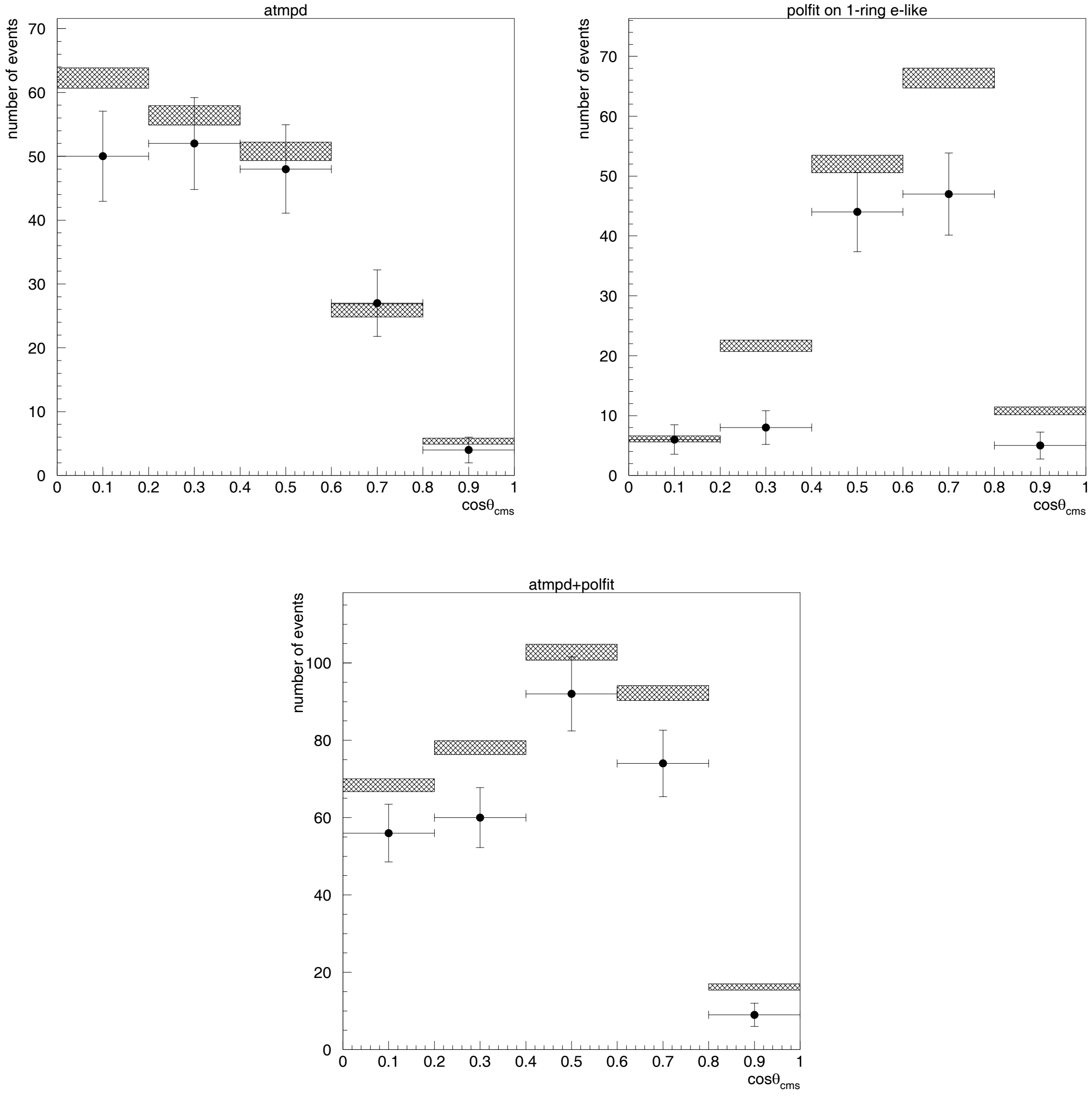


Figure 10.8: $\cos\theta_{\mathrm{cms}}$ distributions for π^0 with $p > 300$ MeV. $\cos\theta_{\mathrm{cms}}$ distributions for π^0 with $p < 300$ MeV. Left plot shows events reconstructed by the standard fitter. Right plot shows the events reconstructed by polfit, which the standard fitter reconstructed as single showering rings. Bottom plot is a sum of the two top plots.

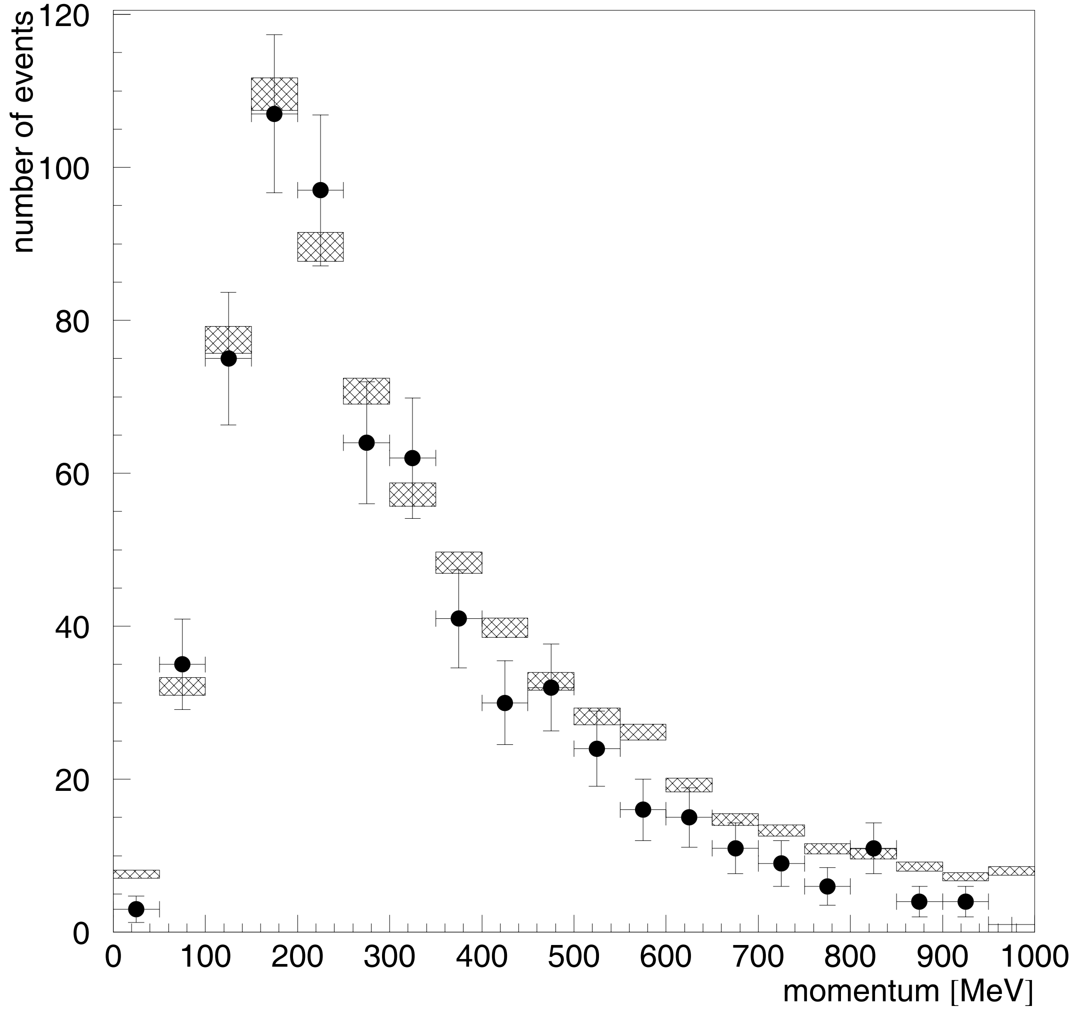


Figure 10.9: Momentum distribution of the π^0 sample (including polfit sample). Points show data, boxes show MC.

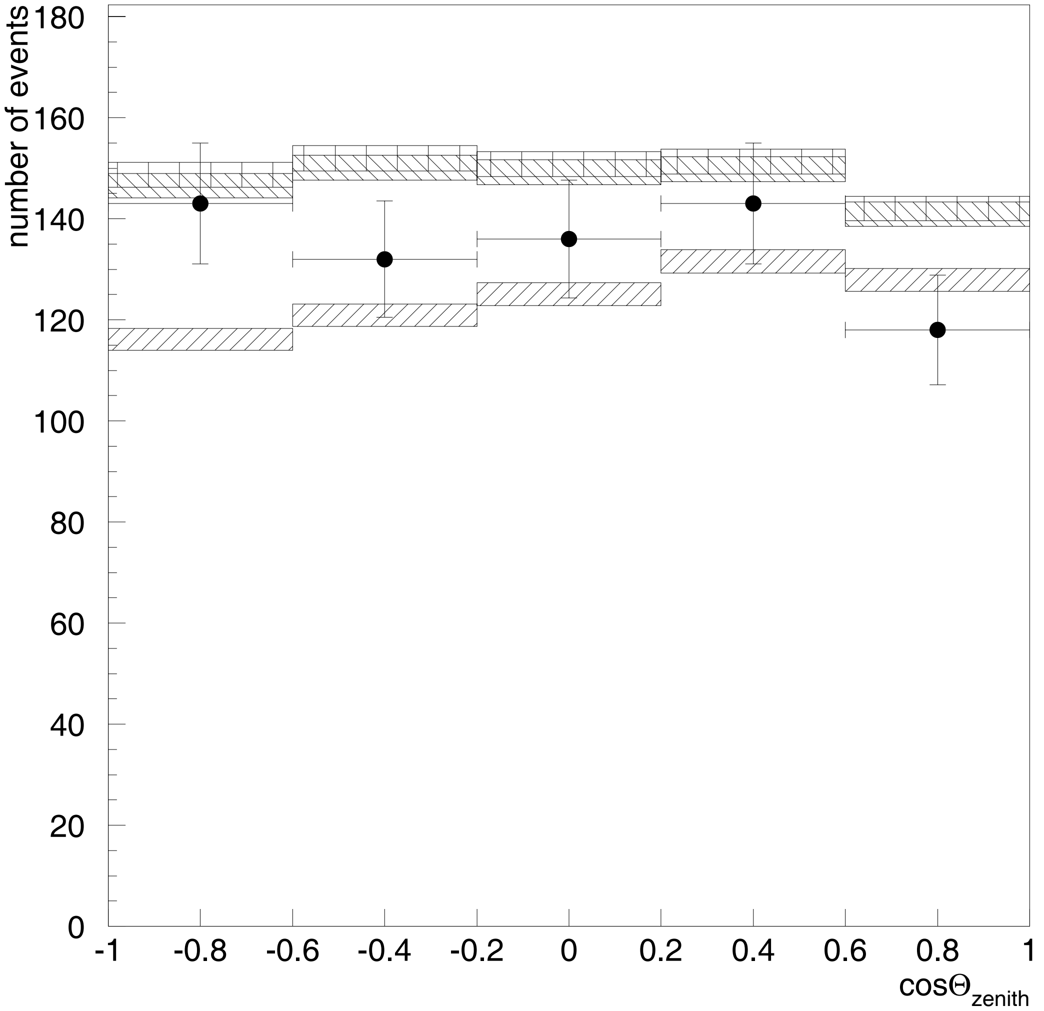


Figure 10.10: Cosine of the zenith angle distribution of the π^0 sample. Points show data, vertically hatched (|||) boxes show MC with no oscillations, boxes hatched with downwards sloping pattern (\\\) show MC with $\nu_\mu - \nu_\tau$ oscillations, boxes hatched with upwards sloping pattern (///) show MC with $\nu_\mu - \nu_s$ oscillations.

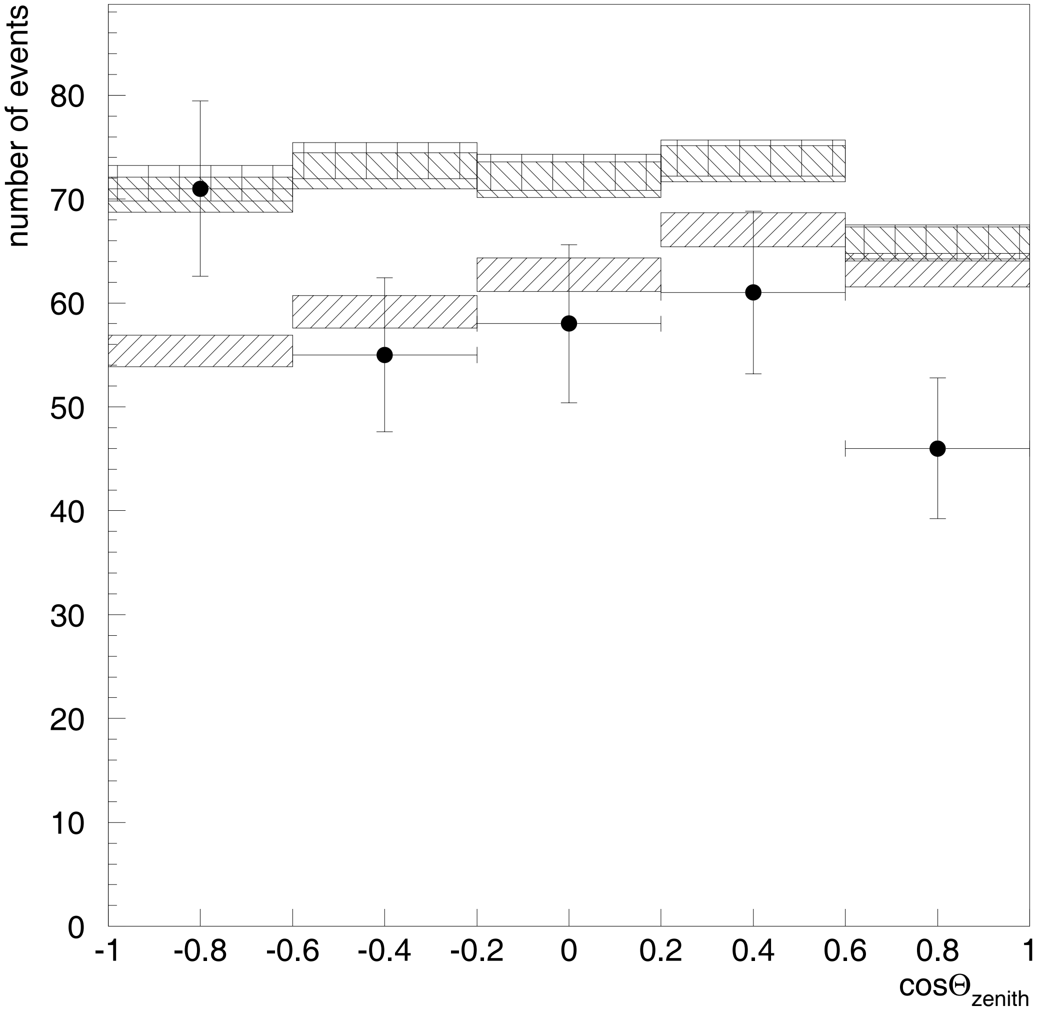


Figure 10.11: Cosine of the zenith angle distribution of the π^0 sample. Only events with momentum greater than 300 MeV are used. Points show data, vertically hatched (|||) boxes show MC with no oscillations, boxes hatched with downwards sloping pattern (\\\) show MC with $\nu_\mu - \nu_\tau$ oscillations, boxes hatched with upwards sloping pattern (///) show MC with $\nu_\mu - \nu_s$ oscillations.

only: up-going and down-going. Then asymmetry quantity can be constructed:

$$A = \frac{d - u}{d + u} , \tag{10.3}$$

where d is the number of down-going events, and u is the number of up-going events. The asymmetry is plotted on Figure 10.12, as a function of the cutoff momentum. The leftmost bin corresponds to no momentum cut performed, just like on Figure 10.10. When calculating the asymmetry, the near-horizontal events with $|\cos\theta_{\text{zenith}}| < 0.2$ (the middle bin on Figures 10.10 and 10.11) are ignored. The Monte Carlo for the two oscillation hypotheses is shown. As expected, the $\nu_\mu - \nu_s$ MC has more positive asymmetry values than the $\nu_\mu - \nu_\tau$ MC, which has the asymmetry near zero. The distinguishing power between the two cases decreases for high momentum cuts because of poor statistics.

At the a priori chosen 300 MeV momentum cut, the asymmetry for $\nu_\mu - \nu_\tau$ is -0.01 ± 0.02, for $\nu_\mu - \nu_s$ it is 0.07 ± 0.02, and for measured data it is -0.08 ± 0.07. The errors are statistical only. This excludes the $\nu_\mu - \nu_s$ hypothesis at 2σ, i.e., 95% confidence level (statistical errors only).

Systematic errors [29] that can affect up-down asymmetry include up-down uncertainty in neutrino flux of the order of 1–2%. Other "asymmetric" systematic errors (reduction and reconstruction, including 0.6% uncertainty of up-down asymmetry of energy calibration) are negligible here.

Systematic errors on things that don't produce asymmetrical distributions can influence the size of the expected asymmetry. Uncertainties in neutrino flux, such as

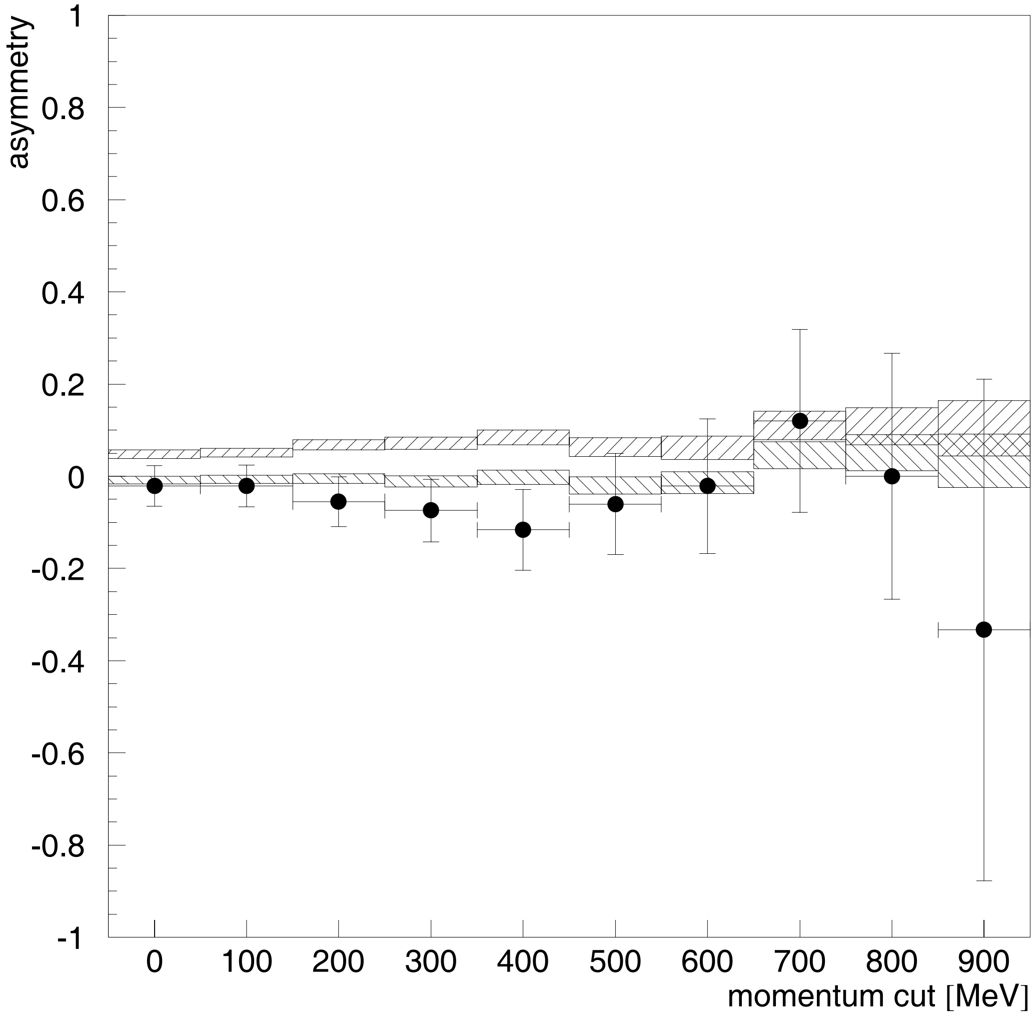


Figure 10.12: Asymmetry $A = (d-u)/(d+u)$ of π^0 zenith angle distribution as a function of momentum cut. Near-horizontal events with $|\cos\theta_{\text{zenith}}| < 0.2$ are not included. Points show data, boxes hatched with downwards sloping pattern (\\\\\\) show MC with $\nu_\mu - \nu_\tau$ oscillations, boxes hatched with upwards sloping pattern (///) show MC with $\nu_\mu - \nu_s$ oscillations.

5% uncertainty in neutrino/antineutrino ratio, don't have a significant effect here. Of uncertainties in neutrino interaction, 20% uncertainty on NC/CC ratio has a direct effect here. This sample consists mostly of NC events (only partially sensitive to asymmetry due to ν_s because of imperfect angular correlation) and some CC events. Even for perfect angular correlation this would translate to effect of about 0.1 out of total 0.7 of the size of the asymmetry, i.e., half of Monte Carlo statistical error and is not significant compared to data statistical error. Imperfect angular correlation reduces this error further. Other neutrino interaction uncertainties include 10% uncertainty on single meson production cross-section, 5% uncertainty on multiple pion production cross-section, and 30% uncertainty on mean free path of hadrons in ^{16}O. Reconstruction errors include few percent uncertainty on ring reconstruction.

Chapter 11

Summary

11.1 Tools

A flexible fitter was developed, called "polfit". The heart of the fitter consists of a likelihood engine. It works by comparing an expected light distribution with an observed light distribution. It can be applied to any kind of events, any configuration of tracks, or even events caused by calibration equipment. The only requirement is that there be an expected light distribution available. This light distribution is automatically computed for Cherenkov light patterns caused by charged tracks. The fitter also contains many general purpose software modules that can be easily used in other projects (light generation, including scattered light, patch generation and manipulation, charge inertia tensor calculation, etc.).

The likelihood engine has been applied to develop a specialized π^0 fitter. It operates by attempting to find a second track in single e-like events. It is shown to reconstruct π^0 events correctly, while most of electron events are reconstructed as non-π^0 events.

Along the way, several useful, general purpose tools were developed.

The need to interface C and Fortran code spurred creation of Fortran to C header translator, called "fh2h", that is integrated in the Super-Kamiokande code.

A general event display program, called "tscan", was developed. It is very flexible, allowing users to view the events in a variety of different ways. It proved to be an invaluable tool for fitter development. An additional, unexpected, benefit of it, turned out to be its role in science popularization.

11.2 Observations

The Super-Kamiokande detector observed atmospheric neutrino oscillations. When this work was started, only single ring events were used. A two ring event sample was analyzed. Two sub-samples were constructed, caused predominantly by charged current interactions of an electron neutrino and a muon neutrino. An up-going muon neutrino disappearance consistent with oscillations is observed in the ν_μ enriched two ring sample. An allowed region of Δm^2 assuming maximal mixing is computed.

The polfit fitter was applied to the atmospheric neutrino sample. It is shown to find π^0 events which were missed by the standard analysis. The zenith angle distribution of the π^0 events has the potential to discriminate between $\nu_\mu - \nu_\tau$ and

$\nu_\mu - \nu_s$ oscillation scenarios. The zenith angle asymmetry is used to test the two possibilities at various minimum momentum cuts. The measured data excludes $\nu_\mu - \nu_s$ hypothesis at 2σ, i.e., 95% confidence level (statistical errors only).

11.3 Future

Next generation neutrino oscillation experiments are planned [37, 38]. In long baseline neutrino oscillation experiments with water Cherenkov detectors, it is crucial to have a good $e - \pi^0$ identification separation. Neutral pion events constitute a major background in search for ν_e appearance in $\nu_\mu \leftrightarrow \nu_e$ oscillation. A specialized fitter, for example, based on the fitter described here, can help with this task.

Bibliography

[1] K. Winter (ed.), *Neutrino Physics*, Cambridge University Press (1991).

[2] C. Sutton, *Spaceship Neutrino*, Cambridge University Press (1992).

[3] A. Franklin, "Experiment in Physics", *The Stanford Encyclopedia of Philosophy* (Spring 2000 Edition), (E. N. Zalta, ed.), `http://plato.stanford.edu/archives/spr2000/entries/physics-experiment/`.

[4] J. Chandwick, Nature **129**, 312 (1932); Proc. Roy. Soc. **A136**, 692 (1932).

[5] E. Fermi, Z. Phys **88**, 161 (1934); Nuovo Cim. **11**, 1 (1934).

[6] R. Peierls, Contemporary Physics **24**, 221 (1983).

[7] F. Reines and C. L. Cowan, Phys. Rev. **92**, 830 (1953).

[8] F. Reines and C. L. Cowan, Nature **178**, 830 (1953).

[9] T. D. Lee and C. N. Yang, Phys. Rev. **104**, 254 (1956).

[10] A. Salam, Nobel Lecture (December 8, 1979).

[11] K. Kodama *et al.*, Phys. Lett. **B504**, 218 (2001).

[12] S. Eidelman *et al.*, Phys. Lett. B **592** (2004).

[13] Ch. Weinheimer *et al.*, Phys. Lett. **B460**, 219 (1999); Phys. Lett. **B464**, 352 (1999) (*erratum*).

[14] M. Lobashev *et al.*, Phys. Lett. **B460**, 227 (1999).

[15] K. A. Assamagan *et al.*, Phys. Rev. **D53**, 6065 (1996).

[16] R. Barate *et al.*, Eur. Phys. J. **C2**, 395 (1998).

[17] N. Spergel *et al.*, Astrophys. J. Supp. **148**, 175 (2003).

[18] D. Davis, Jr., D. S. Harmer, and K. C. Hoffman, Phys. Rev. Lett. **20**, 1205 (1968).

[19] Z. Maki, M. Nakagawa, S. Sakata, Prog. Theor. Phys. **28**, 870 (1962); B. Pontecorvo, Zh. Exp. Theor. Fiz. **53**, 1717 (1967).

[20] G. Fogli, E. Lisi, and G. Scioscia, Phys. Rev. **D52**, 5334 (1995).

[21] L. Wolfenstein, Phys. Rev. **D17**, 2369 (1978).

[22] S. Mikheyev and A. Smirnov, Yad. Fiz. **42**, 1441 (1985).

[23] The Super-Kamiokande Collaboration, Phys. Rev. Lett. **86**, 5651 (2001).

[24] Q. R. Ahmad *et al.*, Phys. Rev. Lett. **87**, 071301 (2001).

[25] G. L. Fogli *et al.*, Phys. Rev. **D66**, 053010 (2002).

[26] Q. R. Ahmad *et al.*, Phys. Rev. Lett. **89**, 011301 (2002).

[27] The KamLAND Collaboration, Phys. Rev. Lett. **90**, 021802 (2003), hep-ex/0212021.

[28] The KamLAND Collaboration, Phys. Rev. Lett. **94**, 081801 (2005), hep-ex/0406035.

[29] The Super-Kamiokande Collaboration, hep-ex/0501064 (submitted to Phys. Rev. **D**) (2005).

[30] The K2K Collaboration, Phys. Rev. Lett. **90**, 041801 (2003), hep-ex/0212007.

[31] The K2K Collaboration, Phys. Rev. Lett. **94**, 081802 (2005), hep-ex/0411038.

[32] The LSND Collaboration, Phys. Rev. **D64**, 112007 (2001).

[33] The KARMEN Collaboration, Phys. Rev. **D65**, 112001 (2002).

[34] E. Church *et al.*, Phys. Rev. **D66**, 013001 (2003).

[35] The BooNE web page: `http://www-boone.fnal.gov/`; hep-ex/0411022 (2004).

[36] The MINOS Experiment and NuMI Beamline web page: `http://www-numi.fnal.gov/`.

[37] NOνA: NuMI Off-Axis ν_e Appearance Experiment web page: `http://www-nova.fnal.gov/`; Letter of Intent, hep-ex/0503053 (2005).

[38] T2K Experiment web page: `http://neutrino.kek.jp/jhfnu/`; Letter of Intent, hep-ex/0106019 (2001).

[39] F. Vissani and A. Y. Smirnov, Phys. Lett. **B432**, 376-382 (1998), hep-ph/9710565.

[40] The Super-Kamiokande Collaboration, Phys. Rev. Lett. **85**, 3999-4003 (2000).

[41] W. W. M. Allison and P. R. S. Wright, "The Physics of Charged Particle Identification", *Experimental Techniques in High Energy Physics*, T. Ferbel ed., Addison-Wesley (1987).

[42] The Super-Kamiokande Collaboration, Nucl. Instrum. Meth. **A501**, 418-462 (2003).

[43] A. Suzuki *et al.*, Nucl. Instrum. Meth. **A329**, 4299 (1993).

[44] Becker-Szendy *et al.*, Nucl. Instrum. Meth. **A324**, 363 (1993).

[45] H. Ikeda *et al.*, Nucl. Instrum. Meth. **A320**, 310 (1992).

[46] KEK Data Acquisition Development Working Group, KEK Report 85-10, (1985).

[47] T. Tanimori *et al.*, IEEE Trans. Nucl. Sci. **NS-36**, 497 (1989).

[48] E. Davides, *Machine Vision: Theory, Algorithms, Practicalities*, Academic Press, San Diego (1997).

[49] Y. Hayato, Nucl. Phys. Proc. Suppl. **112**, 171 (2002).

[50] D. Casper, Nucl. Phys. Proc. Suppl. **112**, 161-170 (2002), hep-ph/0208030.

[51] Y. Shikaze *et al.*, Proc. of the 28th International cosmic Ray Conferences, Vol. 7 p. 4027 (2003).

[52] Neutron monitor page: `http://ulysses.sr.unh.edu/NeutronMonitor/`.

[53] The Super-Kamiokande Collaboration, Phys. Rev. Lett. **82**, 5194 (1999).

[54] M. Honda, T. Kajita, K. Kasahara, and S. Midorikawa, Phys. Rev. **D64**, 053011 (2001), hep-ph/0103328.

[55] M. Honda, T. Kajita, K. Kasahara, and S. Midorikawa (2004), astro-ph/0404457.

[56] G. D. Barr, T. K. Gaisser, P. Lipari, S. Robbins, and T. Stanev, astro-ph/0403630.

[57] G. Battistoni *et al.*, Astropart. Phys. **12**, 315 (2000), hep-ph/9907408.

[58] G. Battistoni, A. Ferrari, T. Montaruli, and P. R. Sala (2003), hep-ph/0305208.

[59] M. Honda, T. Kajita, K. Kasahara, and D. Midorikawa, Phys. Rev. **D52**, 4985 (1995), hep-ph/9503439.

[60] V. Agrawal, T. K. Gaisser, P. Lipari, and T. Stanyev, Phys. Rev. **D53**, 1314 (1996), hep-ph/9509423.

[61] C. H. Llewellyn Smith, Phys. Rept. **3**, 261 (1972).

[62] R. A. Smith ad E. J. Monitzm Nucl. Phys. **B43**, 605 (1972).

[63] D. Rein and L. M. Seghal, Ann. Phys. **133**, 79 (1981).

[64] D. Rein and L. M. Seghal, Nucl. Phys. **B223**, 29 (1983).

[65] D. Rein, Z. Phys. **C35**, 43 (1987).

[66] T. Kitagaki *et al.*, Phys. Rev. **D34**, 2554 (1986).

[67] S. K. Singh, M. J. Vincente-Vacas, and E. Oset, Phys. Lett. **B416**, 23 (1998).

[68] M. Gluck, E. Reya, and A. Vogt, Z. Phys. **C67**, 433 (1995).

[69] The Kamiokande Collaboration, J. Phys. Soc. Jap. **55**, 3786 (1986).

[70] T. Sjostrand, CERN-TH.7112/93 W5035/W5044.

[71] T. Sjostrand, Comput. Phys. Comunn. **82**, 74 (1994).

[72] S. J. Barish *et al.*, Phys. Rev. **D16**, 3103 (1977).

[73] S. Bonetti *et al.*, Nuovo Cim. **A38**, 260 (1977).

[74] S. Ciampolillo *et al.* (Gargemelle Neutrino Propane), Phys. Lett. **B84**, 281 (1979).

[75] N. Armenise *et al.*, Nucl. Phys. **B152**, 365 (1979).

[76] S. V. Belikov *et al.*, Z. Phys. **A320**, 625 (1985).

[77] G. M. Redecky *et al.*, Phys. Rev. **D25**, 1161 (1982).

[78] P. S. Auchincloss *et al.*, Z. Phys. **C48**, 411 (1990).

[79] J. P. Verge *et al.*, Z. Phys. **C35**, 443 (1987).

[80] V. B. Anikeev *et al.*, Z. Phys. **C70**, 39 (1996).

[81] A. S. Volenko *et al.*, Sov. J. Nucl. Phys. **30**, 528 (1979).

[82] D. MacFarlane *et al.*, Z. Phys. **C26**, 1 (1984).

[83] N. Baker *et al.*, Phys. Rev. **D25**, 617 (1982).

[84] R. Woods and D. Saxon, Phys. Rev. **95**, 577 (1954).

[85] L. L. Salcedo, E. Oset, M. J. Vincente-Vacas, and C. Garcia-Recio, Nucl. Phys. **A484**, 557 (1988).

[86] G. Rowe, M. Salomon, and R. H. Landau, Phys. Rev. **C18**, 584 (1978).

[87] D. Ashery *et al.*, Phys. Rev. **C23** 2173 (1981).

[88] R. Brun and F. Carminati, CERN Programming Library Long Writeup W5013 (1993).

[89] C. Zeitnitz and T. A. Gabriel, Nucl. Instrum. Meth. **A349**, 106 (1994).

[90] E. Bracci, CERN/HERA **72-1** (1972).

[91] A. S. Carroll *et al.*, Phys. Rev. **C14**, 635 (1976).

[92] The tscan web page: `http://www.ps.uci.edu/~tomba/sk/tscan/`.

[93] E. W. Weisstein, *CRC Concise Encyclopedia of Mathematics*, 2nd ed., CRC Press LLC (2003); `http://mathworld.wolfram.com/MollweideProjection.html`.

[94] K. Kaneyuki and K. Scholberg, *Neutrino Oscillations*, American Scientist, p. 223 (May-June 1999).

[95] F. Close, M. Marten, and C. Sutton, *The Paricle Odyssey*, title page, and p. 201, Oxford University Press (2002).

[96] I. Semeniuk, *Astronomy and the New Neutrino* Sky & Telescope, p. 44 (September, 2004).

[97] Particle Physics in the UK, Picture of the Week, `http://hepweb.rl.ac.uk/ppUKpics/POW/pr_001206.html`, (6 December 2000).

[98] C. Caso *et al.*, Eur. Phys. J. **C3**, 307-335 (1998); and 1999 off-year partial update for the 2000 edition, `http://pdg.lbl.gov/`.

[99] The LAPACK web page: `http://www.netlib.org/lapack/`.

[100] The fh2h web page: `http://www.ps.uci.edu/~tomba/fh2h/`.

Appendix A

Interfacing C and Fortran Languages

While the algorithms described in in Chapter 8 are implemented in C language, a Fortran interface is also provided. Also, the code described in that chapter, as well as tscan code described in chapter 7, accesses some Fortran subroutines and data structures. These cross-calls are possible by observing Fortran calling conventions in C code.

In order to call a Fortran function from C, one has to call this function in lowercase with an underscore, “_”, appended to the function name. All parameters must be passed by reference, i.e., pointer to the address of a variable should be passed using ampersand “&”. For any string argument, one has to add, after all other arguments, an additional argument that contains string length (passed by value, not by reference).

Fortran common blocks are available as structures with an underscore appended to their names.

In order to make a C function accessible from Fortran, one has to provide a function whose lower-case name ends with an underscore, that takes all parameters by reference, and for each string argument, it takes an integer argument that specifies the string length. In order to make C structure available in Fortran, as a common block, one has to append an underscore to the structure name in lower case.

Note, that while Fortran source code uses reverse order of specifying array indices, the array is laid out in memory in the same way as in C program. Of course, one has to remember, that Fortran, by default, indexes arrays starting from 1 instead of 0.

A.1 Fh2h—Fortran to C Header Translator

If one tries to access, from C, data from some legacy Fortran common block, one has to specify the precise common block layout in C language. This is error-prone, and, whenever the Fortran common block changes, one has to update the C definition. If one forgets to do it, the bugs may be nasty, in a subtle, hard to detect way, that is hard to debug.

The common blocks are usually specified in Fortran include (header) files. What one would like is an automated way to translate these Fortran header files into C include files. If a Fortran header file changes, the build system should detect the change, generate a new C header file, and then "make" dependencies would ensure that C code is recompiled and relinked.

For example, for the following fragment of a Fortran header file,

```
      INTEGER IVAR, IPAR
      REAL RFAR, RPAR
      PARAMETER (IPAR=100,RPAR=3.14)
      INTEGER*4 IARR(IPAR)
      REAL*4 RARR(IPAR)
      COMMON /CMBLK/ IVAR,RVAR,IARR,RARR
```

we would like to have the following code in the C header file:

```
#define IPAR 100
#define RPAR 3.14
struct cmblk_common {
  int   ivar;
  float rvar; /* implicit type */
  int   iarr[IPAR];
  float rarr[IPAR];
} cmblk_;
```

There already exists a program for translation of Fortran code to C code, f2c. However, it is not suitable for header translation, because, for example, parameters are converted to raw numerical constants with no symbolic names available, so, for example, one can't declare an array with the size defined by the parameter. Also, it generates anonymous structures, that are hard to debug with a debugger.

In order to solve this problem, a Fortran to C header translator was developed, fh2h. It is implemented in Perl, as Perl has powerful text processing capabilities. It generates a header file that is compatible with both C and C++, using construct `extern "C" { ... }` bracketed in `#ifdef __cplusplus` conditionals. The header file is protected against multiple inclusions by bracketing the whole file in `#ifdef`

`<name> #define <name> ... #endif`. Numerical and string Fortran parameters are translated to upper-case C preprocessor constants. Fortran types are translated to C types, paying attention to the type size. Translator fh2h handles Fortran arrays with starting index different that 1 (both negative and positive). Fortran data statements are translated into static variables. For example:

```
DATA TEXTS /'text1','text2'/
DATA IMAX /10/
DATA VAL /-9.9988,10E10/
CHARACTER*20 TEXTS(2)
INTEGER IMAX
REAL VAL(2)
```

becomes

```
#ifndef NO_STATIC_DATA
#define TEXTS_MAX_LENGTH (20)
static char   *texts[2] = {"text1", "text2"};
static int    imax = 10;
static float  val[2] = {-9.9988, 10E10};
#endif  /* #ifndef NO_STATIC_DATA */
```

Fh2h preserves Fortran comments, both full-line and in-line kinds. It properly escapes sequences that mask as C comment delimiters, for example, Fortran comment `c...*/...` gets translated to C comment `/* ...* /... */`, i.e., a space is inserted between "*" and "/" in order to prevent premature termination of the comment. Fh2h properly escapes double quotes and backslashes in strings, for example, Fortran string `'"\'` gets translated to C string `"\"\\"`. Fh2h handles Fortran continuation lines. It properly recognizes that Fortran ignores all spaces and case, for example, `In`

`Tegeral he did` becomes `int alhedid;`. The beginning of the generated file contains a comment that explains how this file was generated, and warns against editing the generated file manually.

Translator fh2h is fully integrated into the Super-Kamiokande build system. It is used by other code, besides polfit and tscan.

Fh2h is made freely available for the general public on the World Wide Web [100].